Motto:

"Da steh ich nun, ich armer Tor
Und bin so klug als wie zuvor."

J.W.Goethe FAUST I,Nacht

Meinen Eltern in Verehrung und
Dankbarkeit zugeeignet.

THE UNIVERSITY OF ALBERTA

WAVE FUNCTIONS OF EXCITED ATOMS

by

© UTZ LIEBE

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

DOCTOR OF PHILOSOPHY

DEPARTMENT OF CHEMISTRY

EDMONTON, ALBERTA

SPRING, 1972

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled

WAVE FUNCTIONS OF EXCITED ATOMS

submitted by

UTZ LIEBE

in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

ABSTRACT

The purpose of this work was to test variational principles which have been claimed to be useful for obtaining wavefunctions of ground states and excited states. To this end a series of computer routines has been developed to test these methods for any closed-or open- shell atomic state for up to ten electrons occupying s, p- or d-orbitals. The approximate wavefunction consists of a single configuration. This single configuration is a fully symmetrized sum of slators. Each slator in its turn is expanded as a sum of Slater-type orbital functions. To obtain a self-consistent wavefunction from a starting wavefunction the approach of Hinze and Roothaan to compute corrections to the starting vector is used. The SCF-wavefunctions are used to compute some expectation values related to physical properties.

The variational principles involve the calculation of the expectation value $\langle H \rangle$ which turns out to be a computer-time consuming process. The results show that the computation of this expectation value becomes impractical for larger electronic systems.

The results furthermore confirm that the minimization of delta and delta-tilde leads to wavefunctions which are not useful in computing any physical properties of the state under consideration. The $\mathcal{E}^2/\tilde{\Delta}$ and $\mathcal{E}^2/\Delta$ methods lead to

nearly identical wavefunctions. These wavefunctions show the maximum overlap with the "true" wavefunction in cases where correlation is of minor importance. If the correlation is of major importance then the theoretically expected result is not obtained. If this finding is found to hold true in general then one might employ this method in testing how well a certain wavefunction incorporates correlation effects. It is hoped that the programmes written and the results obtained might serve as a basis for exploring further the nature of excited states and ultimately might lead to to the prediction of physical properties of excited and ground states.

ACKNOWLEDGEMENTS

It is a pleasure for me to express my thankfulness and indebtness to Prof. F. W. Birss, without whom this work would not have been possible. He not only induced me by his exciting and brilliant lecturing to theoretical chemistry but he also provided a never ending source of ideas and suggestions pertaining to this thesis.

I wish to thank Prof. S. Fraga and Prof. S. Huzinaga as well as Dr. A. A. Cantu for helpful discussions.

I am deeply grateful for the help and cooperation provided by the Computing Centre, University of Alberta whose excellent and helpful staff saved uncountable months of fruitless labour. The support of the National Research Council of Canada, in the form of a scholarship, is gratefully acknowledged.

Last, but not least, I want to thank Helen H. J. Liebe for her aid to cast this thesis in its final form and her psychological help that was invaluable in overcoming the tribulations of being a graduate student.

TABLE OF CONTENTS

I. THEORY.

One of the fundamental theorems of quantum mechanics states that each conservative system, i.e. a system whose energy is a constant of the motion, can be represented by the time independent Schroedinger Equation

$$H\Psi = E\Psi \tag{1-1}$$

Although the wavefunction Ψ is not a physical observable, it can be used to derive the physical observables of the system by the relation

$$\langle\hat{A}\rangle = \{\int \Psi^*\hat{A}\Psi \; dv\} / \{\int \Psi^* \Psi \, dv\} \tag{1-2}$$

where $\langle\hat{A}\rangle$ is the mean value of the linear operator $\hat{A}$ which is associated with the dynamical variable $\hat{a}$ according to the postulates of quantum mechanics. This property of the wavefunction has made its determination one of the prime problems in theoretical physics and chemistry.

The solution of the Schroedinger Equation (1-1) in analytical form is possible only in a few cases that can be separated into one-dimensional problems. For any more complicated system one has to resort to approximate methods.

The two most important methods of approximation are

the perturbational and the variational treatments. In perturbation theory one assumes that the Hamiltonian of the system can be separated into two parts

$$H = H^{0} + H^{1} \tag{1-3}$$

For H^{0} a solution of the Schroedinger Equation is known, H^{1} adds a small perturbation.

The variational treatment is derived from the calculus of variations which states in a theorem, that the necessary condition for a function f with continuous first partial derivatives to attain a stationary value is that the first variation δf vanishes for arbitrary changes δx_i in the independent variables x_i. The variational treatment is more general since the solution of the Schroedinger Equation need not be known for part of the Hamiltonian.

Any physical system can exist in several states, a so-called ground state where the system has attained the lowest total energy, and so-called excited states where this is not the case. Excited states play an important role in physics and chemistry, since their observation has furnished us with a wealth of information about physical systems. For example much of the knowledge of such seemingly unrelated things as the composition of stars and

the structure of organic molecules has been obtained by studying excited states.

Excited state wavefunctions could in principle be obtained by varying a trial function ϕ under the constraint that it is orthogonal to all wavefunctions of states of the same symmetry which lie beneath it. (By a theorem of group theory the function is automatically orthogonal to all wavefunctions of states which belong to a different symmetry.) But this is generally impractical since the theorem upon which the above method is based holds rigorously only if the wavefunctions of the lower states are exact.

Excited state wavefunctions could also be obtained by using a theorem of MacDonald (32), applicable to the case where the trial wavefunction is a sum of linearly independent functions, i.e. $\phi = \sum c_i \Gamma_i$. The minimization of the expectation value $\langle \phi | H | \phi \rangle$ leads to an eigenvalue problem $\underset{\sim}{H}\underset{\sim}{c} = \underset{\sim}{E}\underset{\sim}{S}\underset{\sim}{c}$. The theorem states that the eigenvalues E_1, E_2, E_3 , etc. are upper bounds to the true energies W_1, W_2, W_3, etc.. Therefore the eigenvectors c_1, c_2, c_3, etc. can be regarded as approximations to excited state wavefuntions.

Considerable effort has been directed towards obtaining wavefunctions of excited states without imposing the above mentioned constraint.

Of the nonvariational approaches to determine

lower bounds of eigenvalues the method of Lowdin (8), which is based upon perturbation theory, has received a great deal of attention. The method applied to atoms treats the term $H^1 = \sum i<j(1/r^{ij})$ of the Hamiltonian as the perturbation. Not only is it doubtful that for heavier atoms this term is small enough to be treated as a perturbation, but also the computation of the perturbed wavefunctions involves the calculation of the expectation value of the operator $(H^1)^{-1}$, which for any atom with more electrons than Helium leads to presently intractable integrals, e.g. for Li

$$(H^1)^{-1} = (r^{12} * r^{13} * r^{23})/(r^{13} * r^{23} + r^{12} * r^{13} + r^{12} * r^{23})$$

A different method for obtaining bounds, based upon the Rayleigh-Ritz variational method, was suggested in 1934 by Weinstein (6).

If one considers a function

$$\Delta' = \langle\phi|(H-V)^2|\phi\rangle \qquad (1-4)$$

$$1 = \langle\phi|\phi\rangle,$$

V an arbitrary constant

then one can show that for the true eigenvalue W_K of H lying closest to V the relationship

$$V + \sqrt{\Delta} \geqslant W_k \geqslant V - \sqrt{\Delta}$$

holds if W_κ is the only eigenvalue within the range. That V which minimizes Δ' is V=E where $E=\langle\phi|H|\phi\rangle$, which leads to

$$E + \sqrt{\Delta} \geqslant W_k \geqslant E - \sqrt{\Delta}$$

Different ways of obtaining bounds on eigenvalues have been proposed by Temple (24) and Kato (25) and they are mentioned here for completeness only, since they have been investigated as a possible source of excited state wavefunctions by Messmer and Birss (10) and have been found unsatisfactory.

Fraga and Birss (9) have used Weinstein's bounds to suggest a variational procedure which could be used to obtain wavefunctions of excited states.

Messmer (1,10) employed an entirely different approach to obtain wavefunctions. Instead of using the bounds of eigenvalues as a criterion for the "goodness" of a wavefunction, he investigated the quantity:

$$a_\kappa = \langle\phi|\Psi_\kappa\rangle, \quad |\phi\rangle \text{ trial wavefunction;}$$
$$|\Psi_\kappa\rangle \text{ exact wavefunction}$$

This quantity describes the total overlap of the true with

the trial wavefunction. It can be shown that the minimization of Δ does not at all give the best approximation to a Ψ_k that can be obtained from a given trial wavefunction, if one uses the criterion of maximum a_k. Messmer (1,10) then developed a variational scheme for ground and excited states which follows a reasoning of the goodness of approximate wavefunctions in the ground state first given by James and Coolidge (11). Since the work done in this thesis is based essentially upon this scheme the derivation given by Messmer (10) will be repeated in its main parts.

Let ϕ be an approximate normalized wavefunction and $E=\langle\phi|H|\phi\rangle$ its associated energy. Also let

$$\phi = \sum_i \{a_i \Psi_i\} = a_k \Psi_k + \sum_{i\neq k}\{a_i \Psi_i\} \qquad (1\text{-}6)$$

then

$$(\phi - a_k \Psi_k) = \sum_{i\neq k} \{a_i \Psi_i\} \qquad (1\text{-}7)$$

will give the deviation of ϕ from the exact function Ψ_k. A deviation function can then be defined,

$$\phi_\chi = (1-a_k^2)^{-1/2} (\phi - a_k \Psi_k) \qquad (1\text{-}8)$$

where $a_k = \langle\Psi_k|\phi\rangle$. Hence one may write

$$\phi = a_k \Psi_k + a_\chi \phi_\chi \qquad (1\text{-}9)$$

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

where $a_\chi = (1 - a_k^2)^{1/2}$ measures the amount of the deviation function ϕ_χ which appears in ϕ. As criteria of the inaccuracy of ϕ there are :

Q, the root-mean-square error in ϕ

$$Q = \langle \phi - \Psi_k | \phi - \Psi_k \rangle^{1/2} \tag{1-10}$$

$\mathcal{E}$, the energy error

$$\mathcal{E} = E - W_k \tag{1-11}$$

and $\sqrt{\Delta}$, the root-mean-square local energy deviation

$$\sqrt{\Delta} = \langle \phi | (H-E)^2 | \phi \rangle^{1/2}. \tag{1-12}$$

One may also define the quantities:

$$E_\chi = \langle \phi_\chi | H | \phi_\chi \rangle \qquad \mathcal{E}_\chi = E_\chi - W_k$$

$$\Delta_\chi = \langle \phi_\chi | (H-E)^2 | \phi_\chi \rangle. \tag{1-13}$$

If the inequality

$$0 \leq \langle \phi_\chi | [(H-E) - (E_\chi - E)]^2 | \phi_\chi \rangle \tag{1-14}$$

is considered and the integral is expanded in terms of the above defined quantities, one finds

$$0 \leq \langle \phi_\chi | H^2 | \phi_\chi \rangle - 2E_\chi E + E^2 - E_\chi^2 + 2E_\chi E - E^2, \tag{1-15}$$

thus

$$\langle\phi_\chi|H^2|\phi_\chi\rangle - 2E_\chi E + E^2 \geqslant E_\chi^2 - 2E_\chi E + E^2. \quad (1\text{-}16)$$

But the left hand side is merely Δ_χ; hence

$$\Delta_\chi \geqslant (E_\chi - E)^2. \quad (1\text{-}17)$$

From the definition of ε_χ and ε given above, it can be shown that

$$(E_\chi - E)^2 = (\varepsilon_\chi - \varepsilon)^2; \text{ hence } \Delta_\chi \geqslant (\varepsilon_\chi - \varepsilon)^2$$

Thus one may define a quantity K^2,

$$K^2 = \{\Delta/(\varepsilon_\chi - \varepsilon)^2\} \geqslant 1. \quad (1\text{-}18)$$

Now substituting eqn (1-9) into eqns (1-10,1-11,1-12) it follows that

$$Q^2 = 2\{1-a_\kappa\} = 2\{1-(1-a_\chi^2)^{1/2}\} \quad (1\text{-}19)$$

$$\varepsilon = a_\chi^2 \varepsilon_\chi \quad (1\text{-}20)$$

$$\Delta = \varepsilon^2 + a_\chi^2(\Delta_\chi - \varepsilon^2) \quad (1\text{-}21)$$

Eliminating a_χ between eqn (1-19) and eqn (1-20) one finds

$$Q^2 - (1/4)Q^4 = a_\chi^2 = \varepsilon/\varepsilon_\chi \quad (1\text{-}22)$$

or if ϕ is a fairly good approximation to Ψ_K, then

$$Q^2 \cong \varepsilon/\varepsilon_x \qquad (1\text{-}23)$$

Eliminating a_x between eqn (1-20) and eqn (1-21) one obtains

$$\Delta = \varepsilon\{\varepsilon+(\Delta_x-\varepsilon^2)/\varepsilon_x\} \qquad (1\text{-}24)$$

and assuming again that ϕ is a fairly good approximation of Ψ_K, it follows that

$$\Delta/\Delta_x = \varepsilon/\varepsilon_x \text{ or } \Delta \cong \varepsilon\varepsilon_x K^2. \qquad (1\text{-}25)$$

Using eqns (1-23) and (1-25) one obtains

$$Q^2 \cong (\varepsilon^2/\Delta)K^2. \qquad (1\text{-}26)$$

Since $K^2 \gg 1$, one can approximately assume it to be constant and equal to 1 and a minimization of Q^2 will then involve the minimization of ε^2/Δ.

A danger lies in the fact that K^2 could possess a cusp at the point of maximum overlap of ϕ with Ψ_K. But a study of the first ten excited states of hydrogen (12) has shown

that this is not the case, but that K^2 is a slowly varying function in the region where $\phi \to \psi_k$.

In a subsequent paper Choi, Lebeda and Messmer (2) extended the above outlined method and gave an exact formulation.

Defining the quantities

$$\tilde{\Delta} = \langle\phi|(H-W_k)^2|\phi\rangle$$
$$\tilde{\Delta}_x = \langle\phi_x|(H-W_k)^2|\phi_x\rangle \qquad (1\text{-}27)$$

it follows that

$$\tilde{\Delta}/\tilde{\Delta}_x = \langle\phi|(H-W_k)^2|\phi\rangle/\langle\phi_x|(H-W_k)^2|\phi_x\rangle$$

$$= \{\sum m\ |a_m|^2(W_m-W_k)^2 a_x^2\}/$$

$$\{\sum m{\ne}k\ |a_m|^2 (W_m-W_k)^2\} = a_x^2 \qquad (1\text{-}28)$$

and hence one obtains

$$\varepsilon/\varepsilon_x = \tilde{\Delta}/\tilde{\Delta}_x \qquad (1\text{-}29)$$

and therefore

$$1 = (\varepsilon/\varepsilon_x)*(\tilde{\Delta}_x/\hat{\Delta}). \qquad (1\text{-}30)$$

Using the above relationships one may write

$$a_x^2 = \mathcal{E}/\mathcal{E}_x = (\mathcal{E}^2/\tilde{\Delta})*(\tilde{\Delta}_x/\mathcal{E}_x^2). \qquad (1\text{-}31)$$

Now defining $\tilde{K}^2 = \tilde{\Delta}_x/\mathcal{E}^2$ one may write

$$a_x^2 = (\mathcal{E}^2/\tilde{\Delta})K^2 \qquad (1\text{-}32)$$

which is analogous to eqn (1-26) but is an exact relation. To make the connection between eqns (1-26) and (1-32) it is necessary to assume that

$$\tilde{\Delta}/\tilde{\Delta}_x = \Delta/\Delta_x \qquad (1\text{-}33)$$

which is true only in the limit

$$\lim_{\phi \to \Psi_\kappa} (\Delta/\Delta_x) = (\tilde{\Delta}/\tilde{\Delta}_x); \qquad \lim_{\phi \to \Psi_\kappa} E = W_\kappa \qquad (1\text{-}34)$$

then using eqns (1-31) and (1-33) one may write

$$a_x^2 = (\mathcal{E}^2/\Delta)(\Delta_x/\mathcal{E}_x^2) = (\mathcal{E}^2/\Delta)(\Delta_x/[E_x-E]^2) \qquad (1\text{-}35)$$

or

$$a_x^2 = (\mathcal{E}^2/\Delta)K^2 \qquad (1\text{-}36)$$

where eqns (1-36) and (1-26) are the same. Another advant-

age of eqn (1-32) can be seen from the following considerations. If the inequality

$$0 \leq \langle \phi_\chi | [(H-W_\kappa)-(E-W_\kappa)]^2 | \phi_\chi \rangle \qquad (1\text{-}37)$$

is written in terms of the previously defined quantities one obtains

$$\tilde{\Delta}_\chi \geqslant \varepsilon^2 \qquad (1\text{-}38)$$

or

$$1 \leq \tilde{K}^2 = \tilde{\Delta}_\chi / \varepsilon_\chi^2. \qquad (1\text{-}39)$$

Using the above relation one can see from eqn (1-32)

$$a_\chi^2 = 1-a_\kappa^2 \geqslant \varepsilon^2/\tilde{\Delta}. \qquad (1\text{-}40)$$

This lower bound was first derived by James and Coolidge (11).

Let us define a W'_κ as the eigenvalue closest to W_κ ; since the case of degenerate eigenvalues is not considered here, $W'_\kappa \neq W_\kappa$ then

$$\tilde{\Delta}_\chi = \langle \phi_\chi | (H-W_\kappa)^2 | \phi_\chi \rangle$$

$$= a_\chi^{-2} \sum_{m \neq k} |a_m|^2 (W'_k - W_\kappa)^2$$

and therefore

$$\tilde{\Delta}_\chi \geqslant a_\chi^{-2} \sum m \neq k \ |a_m|^2 (W_k' - W_k)^2 = (W_k' - W_k)^2$$

or

$$\tilde{\Delta}_\chi \geqslant (W_k' - W_k)^2 \tag{1-41}$$

From eqns (1-28) and (1-41) it follows that

$$a_\chi^2 = \tilde{\Delta}/\tilde{\Delta}_\chi \leqslant \tilde{\Delta}/(W_k' - W_k)^2. \tag{1-42}$$

Combining this result with eqn (1-40) we obtain

$$\mathcal{E}^2/\tilde{\Delta} \leqslant a_\chi^2 \leqslant \tilde{\Delta}/(W_k' - W_k)^2. \tag{1-43}$$

From the foregoing discussion one can derive variational schemes which will have different properties and will approximate the wavefunction in different parts of the configuration space.

1. MINIMIZATION OF $E = \langle\phi|H|\phi\rangle$.

In general this can only be done, as outlined in the beginning, for the lowest states of a given symmetry. As discussed by Goodisman (13), James and Coolidge (11), this method will minimize "long range errors".

2. MINIMIZATION OF Δ OR $\tilde{\Delta}$.

This method can be used for excited states as well as ground states. In minimizing Δ or $\tilde{\Delta}$ the "local energy error" is minimized (11). The expressions "long range" and "short range" errors, which have been adapted from James and Coolidge (11), warrant some explanation. If one considers the expectation values to be minimized, e.g.

$$\langle H\rangle = \langle\phi|H|\phi\rangle \quad \text{and} \quad \Delta = \langle\phi|(H-E)^2|\phi\rangle$$

and looks at the expanded form of the atomic Hamiltonian (see chapter II) then

$$\langle H\rangle = \langle\phi|[\textstyle\sum_i (-1/2)\nabla_i^2 - Z/r^i] + \sum_{i<j} [1/r^{ij}]|\phi\rangle$$

$$\begin{aligned}\langle H^2\rangle = \langle\phi|[&\textstyle\sum_i (-1/2)\nabla_i^2 - Z/r^i]^2 \\ &+ \textstyle\sum_{i<j} \{[(-1/2)\nabla_i^2 - Z/r^i] * [(-1/2)\nabla_j^2 - Z/r^j]\} \\ &+ \textstyle\sum_{i<j<k} \{[(-1/2)\nabla_i^2 - Z/r^i] * [1/r^{jk}] + [1/r^{ij}] * [1/r^{jk}]\} \\ &+ \textstyle\sum_{i<j<k<l} \{[1/r^{ij}] * [1/r^{kl}]\}|\phi\rangle.\end{aligned}$$

Intuitively one can see that in the expression for Δ an error in the expectation value of $\langle 1/r^{12}\rangle$ will contribute much more than is the case for $\langle H\rangle$. Since $\langle 1/r^{12}\rangle$ will be large when the electrons are close together ("short range")

and small when they are far apart ("long range"), the wavefunctions obtained by the two different methods will therefore reflect this relative importance of $\langle 1/r_{12} \rangle$. Thus, by minimizing E, Δ or $\tilde{\Delta}$, a wavefunction is obtained which has the minimum error in certain regions of configuration space, rather than the minimum error over the whole of configuration space.

3. MINIMIZATION OF $\mathcal{E}^2/\Delta$ OR $\mathcal{E}^2/\tilde{\Delta}$.

These methods are suitable for ground states and excited states. Both methods will provide wavefunctions which show the best overall convergence to the true wavefunction in all parts of configuration space. In the $\mathcal{E}^2/\tilde{\Delta}$ minimization, one has also to insure that $\tilde{\Delta}/(W_k - W_k')^2$ remains "reasonably" small to guarantee that ϕ approaches ψ_k in a least mean-square sense. The smallness of $\tilde{\Delta}/(W_k - W_k')^2$ is of great concern and it has to be decided from particular case to particular case if this criterion is met. In this work all methods are used to approximate various wavefunctions of atoms of the first row of the periodic table.

II. METHODS.

The objective of this work is to find wavefunctions of atoms by minimizing the following quantities:

1) $\langle H \rangle$

2) $\langle (H-E)^2 \rangle$

3) $\langle (H-W_k)^2 \rangle$

4) $\langle H-W_k \rangle^2 / \langle (H-W_k)^2 \rangle$

5) $\langle H-W_k \rangle^2 / \langle (H-E)^2 \rangle$

Under variation all of these methods lead to the same type of equation

$$\delta\langle H \rangle + w\delta\langle H^2 \rangle = 0 \qquad (2\text{-}1)$$

with:

1) $w = 0$

2) $w = -1/2E$

3) $w = -1/2W_k$

4) $w = -\epsilon/2(\tilde{\Delta}+W_k\epsilon)$

5) $w = -\epsilon/2(\Delta+E\epsilon)$

For example:

$$0 = \delta(\epsilon^2/\tilde{\Delta}) = \{(\delta\epsilon^2)\tilde{\Delta}-\epsilon^2\delta\tilde{\Delta}\}/\tilde{\Delta}^2$$

$$0 = 2\epsilon\tilde{\Delta}\delta\epsilon - \epsilon^2\delta\{\langle H^2 \rangle-2W_k\langle H \rangle+W_k^2\}$$

$$= (2\tilde{\Delta}/\epsilon)\delta\langle H \rangle - \delta\langle H^2 \rangle + 2W_k\delta\langle H \rangle$$

$$= \delta\langle H \rangle\{(2\tilde{\Delta}+2W_k\epsilon)/\epsilon\} - \delta\langle H^2 \rangle$$

$$= \delta\langle H \rangle + \{-\epsilon/2(\tilde{\Delta}+W_k\epsilon)\}\delta\langle H^2 \rangle$$

Therefore the problem reduces to finding expressions for $\delta\langle H \rangle$ and $\delta\langle H^2 \rangle$ and combining them in a suitable

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

fashion.

The nonrelativistic Hamiltonian in atomic units (unit of length $\hbar^2/m_e e^2 = 0.52917*10^{-8}$ cm, unit of energy e^2/a_o = 27.210 eV, $\hbar=m_e=e=1$ a.u.) is given by

$$H = \sum_i \{(-1/2)\nabla_i^2 - Z/r^i\} + \sum_{i<j} \{1/r^{ij}\} \quad (2\text{-}2)$$

In terms of 1- and 2-electron operators this is rewritten as

$$H = \sum_i \{h^i\} + \sum_{i<j} \{1/r^{ij}\} \quad (2\text{-}3)$$

From this one obtains in a straightforward manner the expression for the squared Hamiltonian, ordered in 1-,2-,3-, and 4-electron contributions:

$$\begin{aligned} H^2 = & \sum_i \{h_i^2\} \qquad (2\text{-}4) \\ + & \sum_{i<j} \{2h^i h^j + h^i(1/r^{ji}) + (1/r^{ij})h^i + (1/r^{ij})^2\} \\ + & \sum_{i<j<k} 2\{h^i(1/r^{jk}) + h^j(1/r^{ik}) + h^k(1/r^{ij}) \\ & \quad + (1/r^{ij})(1/r^{jk}) + (1/r^{ik})(1/r^{jk}) + (1/r^{ij})(1/r^{ik})\} \\ + & \sum_{i<j<k<l} 2\{(1/r^{ij})(1/r^{ke}) + (1/r^{ik})(1/r^{je}) + (1/r^{ie})(1/r^{jk})\} \end{aligned}$$

Substituting the expression for $\langle H \rangle$ and $\langle H^2 \rangle$ obtained from the orbital approximation into (2-1) leads to the very well known formalism of the Self Consistent Field (SCF)

theory (15), which is dealt with in chapter III.

The only real problem which had to be solved was the formulation of the $\langle H^2 \rangle$-expression, since it contains 3- and 4-electron parts, the contributions of which had not been fully dealt with in the literature. Fraga and Birss (9) give a general expression for Δ, but they do not state how the different coefficients they introduce in this expression can be obtained.

Since it was intended to write a program of general applicability and no obvious way could be seen to find expressions for these coefficients in the general case, a different route was followed.

The expression for the expectation values of any operator O is given by $\langle \phi | O | \phi \rangle$. The wavefunction ϕ is expanded as a sum of slators (slator=Slater determinant), such that this sum is an eigenfunction of the orbital angular momentum operator L^2 and the spin angular momentum operator S^2. Therefore:

$$\phi = \sum_i \{a^i D^i\} \tag{2-5}$$

and the expectation value is rewritten as

$$\langle O \rangle = \sum_{I,J} a^I a^J \langle D^I | O | D^J \rangle \tag{2-6}$$

The evaluation of the expression $\langle D^I|O|D^J\rangle$ for 1- and 2-electron operators has been extensively dealt with in the literature (e.g.16). The evaluation of the 3- and 4-electron parts is more complex (see appendix I), but can be coded for an electronic computer. To find the coefficients a^I in eqn(2-1), a method first suggested by Harris and Schaeffer (17) has been used (appendix III).

The expression to be varied can be expressed as

$$\langle O\rangle = (2l+1)^{-1}(2s+1)^{-1}\sum_{s\geqslant ms\geqslant -s}\sum_{l\geqslant ml\geqslant -l}\langle\phi(ml;ms)|O|\phi(ml;ms)\rangle \qquad (2\text{-}7)$$

or

$$\langle O\rangle = \langle\phi(ml=l;ms=s)|O|\phi(ml=l;ms=s)\rangle \qquad (2\text{-}8)$$

which give the same expectation value.

Imposing the equivalence restriction should assure that expression (2-8) and (2-9) yield the same wavefunction. This has found to be the case for Be $1s^2\ 2s\ 2p\ {}^3P$ and Be $1s^2\ 2p^2\ {}^1D$ and since the expression (2-8) reduces considerably the amount of computation, all expressions to be varied have been expressed for the highest ml- and ms-value only.

As is well known (15,18,19) the matrix of the Lagrangian multipliers in an open shell case cannot be brought into diagonal form by a suitable unitary trans-

formation. To remove this difficulty, various forms of coupling operators have been introduced (15,18,19).

A different approach has been chosen by Hinze and Roothaan (21), where in each iterative step a correction to the Fock matrices is computed. Preliminary studies showed that the formalism of Hinze and Roothaan was numerically more stable and converged faster than the coupling operator method. Therefore this method was chosen in preference over the coupling operator method. The extension of the Hinze-Roothaan formalism to include 3- and 4-electron operators is dealt with in appendix IV.

To conclude this chapter it is appropriate to consider a more mundane aspect: the economics of a computational process involving $\langle H^2 \rangle$ variation using the Hinze-Roothaan formalism. If one employs the Roothaan-expansion method (15) to approximate wavefunctions, then the computation of the 3- and 4-electron operator matrices requires summations of the order n^6 and n^8, where n is the number of basis functions employed.

The number of these summations could in principle be reduced by computing only those matrix elements that cannot be obtained by an exchange of indices from an already computed matrix element. But this proves in general to be very difficult (see appendix II) and therefore the maximum number of summations must usually be carried out. Since the

correction matrices in the Hinze-Roothaan formalism require at least the same amount of computation as the Fock matrices, whereas the coupling operators can be computed by combining the Fock matrices (a relatively short and fast process), it might be worthwhile to trade the fewer iterations of the Hinze-Roothaan method for more, but faster iterations employing the coupling operator method. Since this work was not concerned with a mass production of wavefunctions, but rather with the exploration of various variational schemes, no attempt has been made to shorten the computational process to its lowest limit.

III. THE MATHEMATICAL DEVELOPMENT

The wavefunction Ψ of an atomic state with N electrons of multiplicity ^{2s+1}L is written as a linear combination of slators (to facilitate the understanding of the following development an explicit example is given in appendix V):

$$\Psi(2s+1:1) = \sum_{I} a^{I} D^{I} \tag{3-1}$$

where

$$D^{I} = A \prod_{j=1}^{N} \phi(I;j) \tag{3-2}$$

and

$$(I;j) = n_j^I \; l_j^I \; ml_j^I \; ms_j^I \tag{3-3}$$

expresses that the orbital $\phi(I;j)$ is the j-th orbital in the slator D^I with the associated quantum numbers n_j^I l_j^I ml_j^I ms_j^I.

The antisymmetrizer A is given by

$$A = (N!)^{-1/2} \sum_{p} (-1)^{p} P \tag{3-4}$$

P is a permutation operator belonging to S_n (n=N) and $(-1)^p$ is its associated parity. The summation runs over all the permutations of the group S_n. S_n is the symmetric group of order n (for more details see ref. 23).

The slators are composed of orthonormal orbitals and

therefore

$$\langle\phi(I;j)|\phi(J;k)\rangle = \delta n(I;j)n(J;k)\,\delta l(I;j)l(J;k)$$
$$\delta ml(I;j)ml(J;k)\,\delta ms(I;j)ms(J;k) \quad (3\text{-}5)$$

holds. The coefficients a^I depend on the case in question and are determined as outlined in chapter II and appendix III.

From here on it will be understood that the discussion refers only to the state with the multiplicity ${}^{2S+1}L$ and therefore the superscript ${}^{2S+1}L$ will be dropped.

The expectation value of any operator O is given by:

$$\langle O\rangle = \sum_{I,J} a^I a^J \langle D^I|O|D^J\rangle \quad (3\text{-}6)$$

Substituting the explicit expression of the slator, eqn (3-2), into eqn (3-6) and using the relationship (I-1, 1) yields:

$$\langle O\rangle = \sum_{I,J} a^I a^J \langle \prod_{j=1}^{N}\phi(I;j)|O|\sum_{P}(-1)^P\ P\prod_{k=1}^{N}\phi(J;k)\rangle \quad (3\text{-}7)$$

The operators of interest can in general be written as see(2-4)

$$O = \sum_{p} O^1(p) + \sum_{pq} O^2(p,q) + \sum_{pqr} O^3(p,q,r) + \sum_{pqrs} O^4(p,q,r,s) \quad (3\text{-}8)$$

We now substitute eqn (3-8) into eqn (3-7) and rearrange D^I and D^J to "maximum match", that is, if orbital $\phi(I;j)$ occurs at all in D^J, then it will occur in the same position of D^J as $\phi(I;j)$ occurs in D^I. Thus we obtain:

$$\langle O\rangle = \sum_{I,J} a^I a^J \{ \sum_i \langle\phi(I;i)|O^1|\phi(J;i)\rangle * \nabla N-1$$

$$+ \sum_{i<j} \langle\phi(I;i)\phi(I;j)|O^2|\sum(-1)^p P\phi(J;i)\phi(J;j)\rangle * \nabla N-2$$

$$+ \sum_{i<j<k} \langle\phi(I;i)\phi(I;j)\phi(I;k)|O^3|\sum(-1)^p P\phi(J;i)\phi(J;j)\phi(J;k)\rangle * \nabla N-3$$

$$+ \sum_{i<j<k<e} \langle\phi(I;i)\phi(I;j)\phi(I;k)\phi(I;l)|$$

$$|O^4|\sum(-1)^p P\phi(J;i)\phi(J;j)\phi(J;k)\phi(J;l)\rangle \nabla N-4\} \qquad (3\text{-}9)$$

The summation of the permutation operators runs over all elements of S_2, S_3, S_4, for 2-,3-,4-electron operators, respectively.

The symbol $\nabla N-i$ has the meaning

= 0 if the two slators are not identical in the N-i orbitals not shown in the integral

= 1 otherwise

[illegible]

[illegible]

$$[illegible]$$

$$[illegible]$$

[illegible]

$$[illegible]$$

[illegible]

[illegible] denote the [illegible]

$$[illegible]$$

[illegible]

[illegible]

Expression (3-9) is formidable looking and not easily handled. One can simplify matters considerably if one rewrites the expression for $\langle O \rangle$ as a sum over non-zero integrals by carrying out all the permutations and then integrating over the spinfunctions. This leads to:

$$
\begin{aligned}
\langle O \rangle = {} & \sum_{i=1,I^1} \left[\langle \varphi\{1;i\} | O^1 | \varphi\{2;i\} \rangle * A^i \right] \\
& + \sum_{j=1,I^2} \left[\langle \varphi\{1;j\} \varphi\{2;j\} | O^2 | \varphi\{3;j\} \varphi\{4;j\} \rangle * B^j \right] \\
& + \sum_{k=1,I^3} \left[\langle \varphi\{1;k\} \varphi\{2;k\} \varphi\{3;k\} | O^3 | \varphi\{4;k\} \varphi\{5;k\} \varphi\{6;k\} \rangle * C^k \right] \\
& + \sum_{l=1,I^4} \\
& * \left[\langle \varphi\{1;l\} \varphi\{2;l\} \varphi\{3;l\} \varphi\{4;l\} | O^4 | \varphi\{5;l\} \varphi\{6;l\} \varphi\{7;l\} \varphi\{8;l\} \rangle * D^l \right]
\end{aligned}
\qquad (3\text{-}10)
$$

A^i, B^j, C^k, D^l are constants obtained by summing the various contributions $a^I a^J$ from eqn(3-9). I^1, I^2, I^3, I^4 are the number of non-zero 1-, 2-, 3- and 4-electron integrals respectively. The φ are space-orbitals not including the spin and

$$\{1;i\} = n_i^1 l_i^1 ml_i^1$$

denote the space quantum numbers of the 1-st orbital of the i-th integral.

The advantage of expression (3-10) over other formulations is:

1) It is completely general and holds for any state whatsoever. One obtains such an expression regardless of averaging over all subspecies of a symmetry or just taking into account one particular subspecies.

2) It is easily obtained by a programmed procedure for an electronic computer.

Following the conventional formalism of the Roothaan expansion method, each orbital is expanded in a set of basis functions, which are in our case Slater-type-orbitals (STO).

$$\varphi\{1;i\} = \sum_{p} c(n^{1},1^{1};i;p) * \chi(1^{1},m1^{1};i;p)$$

with

$$\chi(1^{1},m1^{1};i;p) = r^{n_p-1} * e^{-\eta_p r} * N(n_p\ \eta_p) * Y(1^{1},m1^{1};i)$$

$N(n_p\ \eta_p)$ is a normalization constant

$Y(1^{1},m1^{1};i)$ is the spherical harmonic associated with the orbital $\varphi\{1;i\}$

$n_p\ \eta_p$ are constants for each particular basis.

n_p is an integer and

η_p is a constant called the "orbital-exponent" which can be obtained by optimization procedures or application of rules such as the Slater rules (see ref. 27).

It is worthwhile noticing that the expansion coefficients do not depend on the ml-quantum number. This recognizes that the degenerate set differs only in the angular part (see ref. 15).

Since we are using a single configuration approach, each slator is expressed as a set of orbitals that agree in the n- and l-quantum numbers. Any set of n- and l-quantum numbers occurring in one slator must occur in any other slator. (Expression (3-10) holds also for the multiconfiguration case, but the following argument has to be slightly modified to include multiconfiguration formulation).

The expansion coefficients depend only upon the n- and l-quantum numbers. Therefore, the expression (3-10) yields the expanded form (3-11):

$$
\begin{aligned}
\langle O\rangle = {} & \sum_{i=1,l^1} A^i \sum_{p,q} \{ c(n,l;i,p)c(n,l:i,q) \\
& \qquad \langle \chi(n,ml;i,p)|O^1|\chi(n,ml';i,q)\rangle \} \\
+ & \sum_{j=1,l^2} B^j \sum_{p,q,r,s} \{ c(n^1,l^1;j,p)c(n^1,l^1;j,q) \\
& \quad c(n^2,l^2;j,r)*c(n^2,l^2;j,s)*\langle \chi(l^1,ml^1;j,p)\chi(l^2,ml^2;j,r)|O^2| \\
& \qquad |\chi(l^1,ml^{1\prime};j,q)\chi(l^1,ml^{1\prime};j,s)\rangle \} \\
+ & \sum_{k=1,l^3} C^k \sum_{p,q,r,s,t,u} \{ c(n^1,l^1;k,p)c(n^1,l^1;k,q) \\
& \quad c(n^2,l^2;k,r)c(n^2,l^2;k,s)c(n^3,l^3;k,t)c(n^3,l^3;k,u) \\
& \quad \langle \chi(l^1,ml^1;k,p)\chi(l^2,ml^2;k,r)\chi(l^3,ml^3;k,t)|O^3|
\end{aligned}
$$

$$|\chi(l^1, ml^{1\prime}; k, q)\,\chi(l^2, ml^{2\prime}; k, s)\,\chi(l^3, ml^{3\prime}; k, u)\rangle\}$$

$$+\sum_{l=1,I^4} D^l \sum_{p,q,r,s,t,u,v,w} \{c(n^1, l^1; l, p)\,c(n^1, l^1; l, q)$$
$$c(n^2, l^2; l, r)\,c(n^2, l^2; l, s)\,c(n^3, l^3; l, t)$$
$$c(n^3, l^3; l, u)\,c(n^4, l^4; l, v)\,c(n^4, l^4: l, w)$$
$$\langle\chi(l^1, ml^1; l, p)\,\chi(l^2, ml^2; l, r)\,\chi(l^3, ml^3; l, t)\,\chi(l^4, ml^4; l, v)\,|O^4|$$
$$\chi(l^1, ml^{1\prime}; l, q)\,\chi(l^2, ml^{2\prime}; l, s)\,\chi(l^3, ml^{3\prime}; l, u)\,\chi(l^4, ml^{4\prime}; l, w)\rangle\}$$
$$(3\text{-}11)$$

The permutation symbol indicates that the basis functions of the ket must be permuted before the integration is carried out. The prime on the ml-quantum numbers in the ket indicates that they might be different from the ml-quantum numbers in the bra.

To facilitate the writing of expressions we introduce

$$H^{pq}_{\varsigma^1_i} \qquad\qquad K^{(P_1\hat{+}P_2\hat{+}P_3)(p,q;r,s;t\,u)}_{\varsigma^1_k\,\varsigma^2_k\,\varsigma^3_k}$$

$$J^{(P_1\hat{+}P_2)(p,q;r,s)}_{\varsigma^1_j\,\varsigma^2_j} \qquad\qquad L^{(P_1\hat{+}P_2\hat{+}P_3\hat{+}P_4)(p,q;r,s;t,u;v\,w)}_{\varsigma^1_l\,\varsigma^2_l\,\varsigma^3_l\,\varsigma^4_l}$$

where for example

$$J^{P(pq,rs)}_{\varsigma^j_1\,\varsigma^j_2} = \left\langle \chi_p^{l^1_j ml^1_j}\,\chi_r^{l^2_j ml^2_j}\,|O^2|\,\mathcal{P}\,\chi_q^{l^1_j ml^{1\prime}_j}\,\chi_s^{l^2_j ml^{2\prime}_j}\right\rangle$$

with analogous definitions for the other symbols.

To minimize the expectationvalue <O> one subjects the orbitals to a variation and introduces the orthonormality constraint

$$\underset{\sim}{C}^{n^1 l}\ \underset{\sim}{S}^{l}\ \underset{\sim}{C}^{n^2 l} = \delta_{n^1 n^2} \qquad (3\text{-}12)$$

and one obtains the well known set of Hartree-Fock equations:

$$\underset{\sim}{F}^{nl}\ \underset{\sim}{C}^{nl} = \sum_{n' \in l} \underset{\sim}{S}^{l}\ \underset{\sim}{C}^{n'l}\ \underset{\sim}{\epsilon}^{nl;n'l} \qquad (3\text{-}13)$$

where $\epsilon(nl;n'l)$ is the matrix of the Lagrangian multipliers and the F-matrix is given by equation (3-14).

$$
\begin{aligned}
F^{nl}_{pq} = {} & \sum_{i}^{I^1} A^i H^{pq}_{\varsigma^1_i}\, \delta_{n^1_i,n}\, \delta_{l^1_i,l} \\
& + \sum_{j}^{I^2} B^j \sum_{r,s} \left[C_r^{n^2_j l^2_j} C_s^{n\, l}\, \delta_{n^1_j n}\, \delta_{l^1_j l} \mp C_r^{n^1_j l^1_j} C_s^{n_j l_j}\, \delta_{n^2_j n}\, \delta_{l^2_j l} \right] J^{(P_1 \mp P_2)(p,q;r,s)}_{\varsigma^1_j\, \varsigma^2_j} \\
& + \sum_{k}^{I^3} C^k \sum_{r,s} \sum_{t,u} \left[C_r^{n^2_k l^2_k} C_s^{n^2_k l} C_t^{n^3_k l^3_k} C_u^{n^3_k l^3_k}\, \delta_{n^1_k n}\, \delta_{l^1_k l} \right. \\
& \qquad \mp C_r^{n^1_k l^1_k} C_s^{n^1_k l^1_k} C_t^{n^3_k l^3_k} C_u^{n^3_k l^3_k}\, \delta_{n^2_k n}\, \delta_{l^2_k l} \mp C_r^{n^1_k l^1_k} C_s^{n^1_k l^1_k} C_t^{n^2_k l^2_k} C_u^{n^2_k l^2_k}\, \delta_{n^3_k n}\, \delta_{l^3_k l} \Big] \\
& \qquad \times K^{(P_1 \mp P_2 \mp P_3)(p,q;r,s;t,u)}_{\varsigma^1_k\, \varsigma^2_k\, \varsigma^3_k} \\
& + \sum_{l}^{I^4} D^l \sum_{r,s} \sum_{t,u} \sum_{v,w} \left[C_r^{n^2_l l^2_l} C_s^{n^2_l l^2_l} C_t^{n^3_l l^3_l} C_u^{n^3_l l^3_l} C_v^{n^4_l l^4_l} C_w^{n^4_l l^4_l}\, \delta_{n^1_l n}\, \delta_{l^1_l l} \right. \\
& \qquad \mp C_r^{n^1_l l^1_l} C_s^{n^1_l l^1_l} C_t^{n^3_l l^3_l} C_u^{n^3_l l^3_l} C_v^{n^4_l l^4_l} C_w^{n^4_l l^4_l}\, \delta_{n^2_l n}\, \delta_{l^2_l l} \\
& \qquad \mp C_r^{n^1_l l^1_l} C_s^{n^1_l l^1_l} C_t^{n^2_l l^2_l} C_u^{n^2_l l^2_l} C_v^{n^4_l l^4_l} C_w^{n^4_l l^4_l}\, \delta_{n^3_l n}\, \delta_{l^3_l l} \\
& \qquad \mp C_r^{n^1_l l^1_l} C_s^{n^1_l l^1_l} C_t^{n^2_l l^2_l} C_u^{n^2_l l^2_l} C_v^{n^3_l l^3_l} C_w^{n^3_l l^3_l}\, \delta_{n^4_l n}\, \delta_{l^4_l l} \Big] \\
& \qquad \times L^{(P_1 \mp P_2 \mp P_3 \mp P_4)(p,q;r,s;t,u;v,w)}_{\varsigma^1_l\, \varsigma^2_l\, \varsigma^3_l\, \varsigma^4_l} \qquad (3\text{-}14)
\end{aligned}
$$

The addition symbol $\hat{+}$ used in eqn (3-14) signifies that each of the expansion coefficient products in preceeding square brackets is associated with a different ordering of superscripts on the integral symbols.

The above equations are used in the conventional SCF iterative procedure until self-consistency is obtained.

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

IV. RESULTS AND DISCUSSION.

IV.1. The possibilities of the program.

The program was set up with the aim of providing the highest flexibility and generality possible. Since the computer is of finite size, certain limits have to be defined from the beginning. These limits were chosen so as to compromise between computer efficiency and desired generality. The program in its present form handles only single configurations, but it should not prove too difficult to reformulate certain parts of the program to include the multiconfiguration case.

It was felt that the Russel-Saunders coupling does not describe the true state of open shells of many electrons and therefore an arbitrary maximum of ten electron systems was chosen. This limit seemed also to be sensible for the reason that it allows complete coverage the first row of the periodic system of elements.

Another choice had to made with respect to the maximum number of slators admissible. Again an arbitrary maximum, this0time of up to 52 slators, was chosen after some experimentation and considerations of the findings of Harris and Schaeffer (17). In all of the work carried out these limits were never approached and it is felt that they

actually could be lowered.

For computational purposes a selection was made of the orbitals admissible in the configurations to be computed. This selection could be easily extended if the need should arise and is in the current form of the program:

s-orbitals: 1s 2s 3s 4s

p-orbitals: 2p 3p 4p 5p

d-orbitals: 3d 4d

A more severe restriction is imposed upon the number of basis functions by the finite size and finite speed of the computer. If one stores the 3-electron integrals in a 6-dimensional array then one uses n^6 8-byte storage locations which means 125 kilo bytes (K) for five basis-functions and 373 K for six basis functions and 2097 K for eight basis-functions. (In this chapter confusion might arise over the meaning of the word 'integral'. The word integral is used in two different ways: the first way it is used denotes integrals over orbitals, the second way it is used is the description of integrals over Slater type functions. To avoid misunderstanding, the first type will be written in capital letters, i.e. INTEGRAL whereas the second type will be written in lower case letters.) The reduction of the number of storage locations was attempted by storing only the distinctly different integrals, but this proved to be a very computer-

time consuming and elaborate process, since the number of distinct integrals depends upon the operator P in expression (3-9) and a different routine would have to be written for each case. On the other hand work by Roothaan and Bagus (26) has shown that for the first row elements an optimized set of five s-basis functions suffices to give an adequate SCF-wavefunction. This finding together with the above mentioned difficulty has led to a limitation of a maximum number of five expansion functions. To alter this limit the structure of the whole program, especially the part where the 3-electron matrices are set up, would need to be changed. It is felt that at the present moment such a change is not necessary. To obtain the best possible wavefunction, with the limited bases set, a routine has been set up which allows optimization of the orbital exponents of each expansion function following the "brute force" method suggested by Roothaan and Bagus (26). This optimization routine allows the exponents to be optimized with respect to the various quantities being minimized.

[illegible]

IV. 2. An overview of the states for which calculations have been undertaken.

During the development of the program a variety of states have been computed. Not all the calculations will be discussed in detail. Either the results for these states have been obtained previously or a detailed discussion of these states would not add considerably to an understanding or their methods employed and of the characteristics. Table (IV-1) lists all of the states for which calculations have been carried out and for which a self-consistent wavefunction was obtained.

The symbols n, η and c(nl) in tables IV-2 to IV-14 are explained on page 26.

TABLE IV-1

OVERVIEW OF CALCULATIONS

ATOM	CONFIGURATION	STATE	METHOD*
He	$1s^2$	^1S	1,2,3,4,5
	$1s\ 2s$	$^1S, ^3S$	1, 4,5
	$1s\ 3s$	$^1S, ^3S$	4,5
	$1s\ 2p$	$^1P, ^3P$	1,2,3,4,5
	$1s\ 3p$	$^1P, ^3P$	1,2,3,4,5
	$1s\ 4p$	$^1P, ^3P$	2,3,4,5
	$2s\ 2p$	^1P	4
Li^+	$1s^2$	^1S	1,2,3,4,5
Li	$1s^2 2s$	^2S	1,2,3,4,5
	$1s^2 3s$	^2S	2,3,4,5
	$1s^2 2p$	^2P	1,2,3,4,5
	$1s^2 3p$	^2P	2,3,4,5,6
Be	$1s^2 2s^2$	^1S	1,2,3,4,5
C	$1s^2 2s^2 2p^2$	$^3P, ^1S, ^1D$	1

* 1: $\langle H \rangle$-minimization

2: $\langle (H-E)^2 \rangle$-minimization (Δ-minimization)

3: $\langle (H-W)^2 \rangle$-minimization ($\tilde{\Delta}$-minimization)

4: $\epsilon^2/\tilde{\Delta}$-minimization

5: ϵ^2/Δ-minimization

6: $t*\epsilon^2+\tilde{\Delta}$ -minimization, where t is a parameter

IV. 3. Computing times.

The most severe limitation of any variational method involving the operator H^2 lies in the large amount of computing time required. It should be well born in mind that the following discussion of computing times in the case of the ground state of Be is based upon the very general program where not every possible way of shortening the execution time has been exploited. At the end of this chapter an attempt is made to calculate an upper limit beyond which any variational method involving the H^2-operator would become impractical with the currently available computers.

The expectation value $\langle H^2 \rangle$ for the ground state of Be $1s^2 2s^2$ ^1S consists of two 1-electron operator INTEGRALS, four 2-electron operator INTEGRALS, four 3-electron operator INTEGRALS and three 4-electron operator INTEGRALS. The s-orbitals are expanded into five Slater-type-orbitals (STO). The computation of the 1-electron operator matrices does not consume more than 0.001% of the total computing time (usually far less) and therefore their contribution will be neglected. In each computation one has two main parts:

a) The computation of the integrals between STO's.

b) The computation of the matrix elements of the Fock

matrices using the computed integrals between STO's.

Part a) is only done once in each calculation, part b) is repeated until self-consistency within a certain degree of accuracy is obtained. (With the IBM 360/67 computer at the University of Alberta Computing Centre 10^{-14} is the highest sensible accuracy which can be obtained between two succesive iterations, but one usually stops the calculation when an accuracy of 10^{-4} to 10^{-8} has been obtained).

Since the wavefunction for Be can be approximated using only s-orbitals, s-type STO integrals only are computed.

For five basis functions this requires the computation of 5^4 = 625 $(1/r^{12})$, 625 $(1/r^{12})^2$, 3*625 (h^1*1/r^{12}) integrals between STO's for the 2-electron operators and $6*5^6$ = 93,750 $(r^{12}r^{23})^{-1}$ and $3*5^6$ =46,875 (h^1*1/r^{23}) integrals between STO's for the 3-electron operators. No integrals between STO's for 4-electron operators have to be computed, since these integrals can be obtained by multiplying the appropriate $(1/r^{12})$ STO integrals.

The computation of the 3125 2-electron operator STO integrals took 11.667 sec, the computation of the 140, 625 3-electron operator STO integrals consumed 1,068.171 sec. (All times are Central Processing Unit times.)

As a rule these integrals are transferred from the temporary magnetic disk storage to magnetic tape and can

[illegible]

then be used in the calculation of different variational schemes.

To set up the Fock matrices for 2-electron operators one has to compute 8*25 matrix elements which involves 4*4*625 summations. This step takes only 0.759 sec. In all cases the computation of the 2-electron operator matrices required usually a negligible amount of time.

The 3-electron operator matrices involve a far greater number of summations. For each INTEGRAL 5^6 = 15,625 summations are carried out. Since one has also to compute the correction matrices of the Hinze-Roothaan formalism, the total number of summations increases to 4*15*15,625 = 937,000 summations using 33.565 sec.

The number of computations increases sharply for the 4-electron operator matrices. For the three INTEGRALS in the present case it includes 3*28*5^8 = 32,812,500 summations using 1,314.394 sec.

With the present general program the minimization of $\epsilon^2/\tilde{\Delta}$ for the Be $1s^2 2s^2\ {}^1S$ case employing five basis functions uses approximately 22 min per iteration and an additional 18 min to calculate the STO integrals. If the results of the computation using two basis functions can be extrapolated, four iterations should suffice to achieve convergence, which would yield an overall time of 106 min. It is felt that these requirements upon the computing facil-

ities are excessive.

How would one fare using all possible ways to cut down the required time? As is clear from the foregoing discussion, not much will be gained improving the computation of the 1- or 2-electron operator matrices. If one abandons the Hinze-Roothaan formalism and uses coupling operator methods, one would roughly diminish the time for computing 3-electron operator matrices by a factor of five and 4-electron operator matrices by a factor of seven, diminishing the total time for each iteration from 1349 sec to 195 sec. Further, computing only the necessary elements would reduce the number of computations for the s-type integrals in the Be ground state case from $(n^2)^3$ and $(n^2)^4$ to $[n*(n+1)/2]^3$ and $[n*(n+1)/2]^4$, reducing the computation time for the 3-electron operator matrices by a factor of five and of the 4-electron operator matrices by a factor of eight, decreasing the total for each iteration to about 30 sec. Computing only those STO integrals that are unique, one would be able to cut the computing time of the integrals to about 9 min, and the computation of the ground state of Be would require a total of 11 min. This time seems to be the lower limit which could be reached.

It is interesting to extrapolate these times to the case of C $1s^2 2s^2 2p^2$ ^1S. The expression for $\langle H^2 \rangle$ in this

configuration consists of, other than the 1- and 2-electron operator INTEGRALS, the 51 3-electron operator INTEGRALS and 100 4-electron operator INTEGRALS. For this state one should use five basis functions for both the s- and p-orbitals. One can roughly estimate that half of the INTEGRALS would be highly symmetrical allowing the same saving as in the case of Be $1s^2 2s^2$. The other half would require n^6 and n^8 summations for the 3- and 4-electron operator INTEGRALS respectively. Based upon these estimates the 3-electron operator matrices would require 47 sec and the 4-electron operator matrices 2920 sec per iteration. Thus each iteration would approximately require 50 min.

A conclusion from the foregoing sections is that any variational principle using the $\langle H^2 \rangle$ expression is impractical with present day computer technology for systems larger than Boron or Carbon.

IV.4 Singly Excited States of Helium

In order to test the program the ^1P, ^3P states of He 1s np n=2,3 and 4 were computed and the results compared with the values obtained by Messmer (10). Since these results are essentially identical they are not listed here.

A little bit more complex is the computation of the ^1S and ^3S state for the He 1s ns n=2,3 since two open shells of the same symmetry are present. Tables IV-2 and IV-3 show the results of these computations. It should be noted that the orbital exponents are not optimized but are obtained by applying Slater's rules. The energy of these states computed by Davidson (31) are closer to the experimental energy than those of tables IV-2 and IV-3. This result is to be expected as discussed by Messmer (10).

TABLE IV-2

He ^1S-states $\epsilon^2/\tilde{\Delta}$-minimization

CONFIGURATION	1s 2s	1s 3s
EXP.ENERGY	-2.14572	-2.06104
CALC.ENERGY	-2.167767	-2.014324
$\langle H^2 \rangle$	4.716966	4.064116
EPSILON	0.022	0.047
DELTA	0.017751	0.01064
DELTA-TILDE	0.018237	0.01291
$\langle 1/r \rangle$	2.345	2.209
$\langle r \rangle$	5.092	12.810
$\langle r^2 \rangle$	22.908	191.681

WAVEFUNCTION

CONFIG. :		1s 2s				1s 3s	
n	η	c(1s)	c(2s)	n	η	c(1s)	c(3s)
1	1.7	1.28464	-0.32975	1	1.7	1.30372	0.12457
2	1.7	-0.42489	0.03875	2	1.7	-0.36615	0.07879
2	0.6	0.36863	0.89374	2	0.4	-0.12466	-4.03900
3	0.6	-0.39368	0.27513	3	0.4	0.38832	8.96681
4	0.6	-0.16346	-0.15221	4	0.4	-0.27712	-5.55659

[illegible]

[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]

[illegible]

[illegible]			[illegible]		[illegible]
[illegible]	[illegible]		[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

TABLE IV-3

He ^3S-states $\epsilon^2/\tilde{\Delta}$-minimization

CONFIGURATION	1s 2s	1s 3s
EXP.ENERGY	-2.17498	-2.06845
CALC.ENERGY	-2.169550	-2.073443
$\langle H^2 \rangle$	4.83565	4.082679
EPSILON	0.005	0.0049
DELTA	0.128702	0.021646
DELTA-TILDE	0.128731	0.021648
$\langle 1/r \rangle$	2.304	2.251
$\langle r \rangle$	5.113	10.092
$\langle r^2 \rangle$	23.208	124.771

WAVEFUNCTION

CONFIG. :	1s 2s				1s 3s		
n	η	c(1s)	c(2s)	n	η	c(1s)	c(3s)
1	1.7	1.28262	-0.06240	1	1.7	1.30888	0.14552
2	1.7	-0.33736	-0.09171	2	1.7	-0.38163	0.16194
2	0.6	-0.00212	1.01654	2	0.4	0.03540	-4.49143
3	0.6	-0.14865	0.09985	3	0.4	-0.02394	9.34964
3	0.6	0.06388	-0.06865	4	0.4	0.01067	-5.20429

[illegible]

A Doubly Excited State of Helium

After having replicated the calculations for the excited states 1s np (n=2 to 4) and computed some other singly excited states, (shown in tables IV-2,3), it seemed to be challenging to attempt a more difficult task, namely He 2s2p ^1P which is reported in the literature by Madden and Codling (25) to have an absorption at 206.2 Å (0.6768 a.u.). Some experimentation was necessary before orbital exponents were found that gave a satisfactory convergence. In the first trial the s-orbital was expanded in 4 STO (n=1 to 4) and the p-orbital in 4 STO (n=2 to 5) with an orbital exponent η = 0.85 for all eight basis functions. This starting wavefunction could not be brought to convergence. The same problem arose when the orbital exponent was changed to 0.5 to allow for greater diffuseness. From these two runs the impression was gained that a shifting of the weight of the wavefunction between 2s and 3s STO was mainly responsible for the nonconvergence.

Therefore the following STO's

s-orbital: 1 0.850, 2 0.850, 3 0.500, 4 0.500

p-orbital: 2 0.885, 3 0.885, 4 0.885, 5 0.885

were tried and led to a rapid convergence (5 iterations) with an energy of -0.655 a.u. This wavefunction was then used as the starting point of an optimization run, which yielded the results in Table IV-4. The energy differ-

ence of 0.0195 a.u. is in fact very satisfactory, since the difference between experimental and Hartree-Fock energy for the ground state is 0.041 a.u. From a priori considerations one would expect the Hartree-Fock energy of the 2s 2p state to be closer to the experimental energy, since the correlation between a 2s and a 2p electron should be much smaller than the correlation between two 1s electrons. Another indication for the "correctness" of the wavefunction are the values for $\langle 1/r \rangle$, $\langle r \rangle$ and $\langle r^2 \rangle$ which give average distances of the electrons from the nucleus. The value of 7.054 a.u. is about 0.8 a.u. larger than the corresponding value for the Be ground state, which indicates a rather diffuse electron "cloud". This diffuseness0again is to be expected from a priori considerations. It is believed that the wavefunction of Table IV-4 represents the He 2s 2p ^1P state as well as can be expected in the context of the Hartree-Fock orbital expansion approximation.

TABLE IV-4

He 2s 2p ^1P $\epsilon^2/\tilde{\Delta}$-minimization

EXP.ENERGY	-0.67684
CALC.ENERGY	-0.657306
$\langle H^2 \rangle$	0.475883
EPSILON	0.019531
DELTA	0.043831
DELTA-TILDE	0.044212
$\langle 1/r \rangle$	0.795
$\langle r \rangle$	7.054
$\langle r^2 \rangle$	32.013

WAVEFUNCTION

2s			2p		
n	η	c(2s)	n	η	c(2p)
1	0.845	-0.63350	2	0.881	1.38753
2	0.907	1.02653	3	0.929	-0.52650
3	0.569	0.58179	4	0.885	0.14679
4	0.467	-0.06362	5	0.885	-0.15918

IV-5 Computation of the Li $1s^2np$ ^2P states.

To demonstrate the utility and versatility of the $\epsilon^2/\tilde{\Delta}$ method, it was intended to compute the series Li $1s^22p$ ^2P, Li $1s^23p$ ^2P and Li $1s^24p$ ^2P. The calculation of the Li $1s^22p$ ^2P is shown in Table IV-5. The orbital exponent for the p-orbitals was derived from Slater's rules (27). The orbitals exponents for the s-orbitals were taken from Huzinaga's tables (19).

But difficulties were encountered when it was tried to extend the same approach to the state Li $1s^23p$ ^2P where again the orbital exponents for the p-orbitals were derived fromOSlater's rules. The result obtained (table IV-7) looked very much like that obtained for the lowest ^2P state (table IV-5). To confirm this interpretation the expectation value <H> was minimized using the same starting vector and orbital exponents (table IV-6). It is evident that $\epsilon^2/\tilde{\Delta}$ converged towards the ground state and not the excited state.

TABLE IV-5

Li $1s^2 2p$ ^2P $\epsilon^2/\tilde{\Delta}$-minimization

EXP.ENERGY	-7.40987
CALC.ENERGY	-7.363932
$\langle H^2 \rangle$	55.78432
EPSILON	0.046
DELTA	1.556814
DELTA-TILDE	1.558935
$\langle 1/r \rangle$	5.678
$\langle r \rangle$	5.929
$\langle r^2 \rangle$	28.505

WAVEFUNCTION

	1s			2p	
n	η	c(1s)	n	η	c(2p)
1	2.482	0.86898	2	0.650	0.74159
1	4.687	0.13107	3	0.650	-0.12270
2	0.672	-0.00196	4	0.650	0.53731
2	1.975	0.02246	5	0.650	-0.09639

TABLE IV-6

Li $1s^2 3p$ ^2P $\langle H \rangle$ -minimization

EXP.ENERGY	-7.33687
CALC.ENERGY	-7.364697
EPSILON	0.0376
$\langle 1/r \rangle$	5.633
$\langle r \rangle$	5.960
$\langle r^2 \rangle$	29.012

WAVEFUNCTION

1s			3p		
n	η	c(1s)	n	η	c(3p)
1	2.482	0.89293	2	0.430	1.75996
1	4.687	0.11263	3	0.430	-1.39675
2	0.672	0.00073	4	0.430	0.88507
2	1.975	-0.01137	5	0.430	-0.29975

TABLE IV-7

Li $1s^2 3p$ ^2P $\epsilon^2/\tilde{\Delta}$-minimization

EXP.ENERGY	-7.33687
CALC.ENERGY	-7.364627
$\langle H^2 \rangle$	55.726767
EPSILON	0.0277
DELTA	1.489023
DELTA-TILDE	1.489793
$\langle 1/r \rangle$	5.649
$\langle r \rangle$	5.953
$\langle r^2 \rangle$	28.965

WAVEFUNCTION

1s			3p		
n	η	c(1s)	n	η	c(3p)
1	2.482	0.88896	2	0.430	1.76475
1	4.687	0.11674	3	0.430	-1.40734
2	0.672	0.00036	4	0.430	0.90346
2	1.975	0.01190	5	0.430	-0.30247

[illegible]

[illegible]

[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]

Subsequently a large number of calculations have been carried out in an attempt to obtain a wavefunction which could with certainty be assigned to a definite excited state.

The first attempt involved changing various orbital exponents and using shorter expansions. These trials led to wavefunctions that possessed various different minima, but due to the size of the values of $\tilde{\Delta}$ and $\tilde{\Delta}/(W_{\kappa} - W_{\kappa}^{!})^2$ no assignment to a definite configuration could be made. Then it was believed that it could be useful to combine the advantage of a small delta with a small epsilon by minimizing the expression $\tilde{\Delta} + t*\epsilon^2$ with various values of t. Again the same behaviour as in the previous attempts was observed: convergence towards different minima which could not be associated unequivocally with a certain configuration. After this method had been exhaustively tried it was attempted to annihilate any effect the Hinze-Roothaan formalism might bear upon the direction of convergence and a normal Jacobi-diagonalization was carried out upon the F-matrices. This is possible for the configuration discussed since it consists of maximally one open shell for each symmetry representation and the matrix of the Lagrangian multipliers can therefore be diagonalized by a unitary transformation. Some problems arose here, because it is not a priori evident which eigenvector should be

selected as the vector for the next approximation. This was solved by having all vectors displayed on a computer terminal and selecting the ones which seemed to possess the right eigenvalues. With this method a convergence problem was encountered since small changes in the s-orbitals led to large changes in the p-orbital. To overcome this difficulty various "frozen" orbital approaches were used and convergence was reached. Again the resulting wave-function could be made to approach various minima and no clear designation could be made of which wavefunction belongs to which state.

A final approach then utilized hydrogenic p-orbitals with various orbital exponents and the $\epsilon^2/\tilde{\Delta}$ method to approach the excited state. Since a compilation of all these data would enlarge this thesis unreasonably, only the main results and the conclusions from all these computations will be given in section IV-8.

A finding which was encountered in every one of the computations was a variety of minima in the energy surface which could be approached by the methods. The exact value of these minima changed with the orbital exponents used, but mostly two to three minima could be clearly separated by using various kinds of starting vectors.

A striking example of the sensitivity of the method to small changes in the orbital exponents or starting vectors

was provided when hydrogenic p-orbitals were employed which were either left unchanged during the iterations ("frozen" p-orbital) or were subjected to the minimization procedure ("floating" p-orbital). Table IV-8 shows the results which were obtained for various orbital exponents for the p-orbitals. By using the vectors obtained from a self-consistency run with "floating" orbitals as starting vectors for a self-consistency run with frozen p-orbitals, the energies which were before between 6.9 and 7.1 a.u. then fell into the range 7.2 to 7.36 a.u.. These fairly widely separated energies seem to suggest that the one expansion (i.e. STO basis functions with the same orbital exponents) can serve as an approximate wavefunction for several different configurations.

TABLE IV-8

Li $1s^2 3p\ ^2P$ $\epsilon^2/\tilde{\Delta}$ -minimization

s-orbital exponents: 1,4.5; 1,3.4; 1,2.4; 1,1.6; 1,0.6

p-orbital exponent (hydrogenic orbital)	ENERGY p-fixed	ENERGY p-floating
0.20	-7.131923	-7.336639
0.25	-7.348105	-7.348840
0.30	-7.076623	-7.356811
0.35	-7.375001	-7.361310
0.40	-7.018950	-7.363182
0.50	-6.958879	no convergence
0.60	-7.285780	no convergence

IV.6 Comparison of the Variational Methods Using Be $1s^2 2s^2$

To compare the different variational principles the ground state of Be $1s^2 2s^2$ ^1S was computed, using two STO expansions. This very truncated basis set was used in order to save computing time, since the orbital exponents were fully optimized with respect to the quantity being minimized. It is not expected that the obtained wave-functions are of high quality but it is believed that they are sufficient to demonstrate the differences which exist between the different variational methods.

On the other hand the computation of a smaller system such as Li or He was not considered, in order to demonstrate the correctness of the 4-electron operator matrix routines and to see if there is a qualitative difference in going from a 3-electron system (Li) to a 4-electron system.

The same relationships which show up in this case have been found to hold true for all other states computed, whether the orbital exponents were optimized or not. Tables IV-9 to IV-13 show the results obtained. Computing times for0the full optimization process were in between 15 - 20 minutes per state.

TABLE IV -9

Be $1s^2\ 2s^2$ ^1S $\langle H \rangle$ -minimization

EXP.ENERGY	-14.66785	
CALC.ENERGY	-14.556739	
$\langle H^2 \rangle$	215.556370	*
EPSILON	- 0.111107	
DELTA	3.657720	*
DELTA-TILDE	3.670033	*
$\langle 1/r \rangle$	8.404	
$\langle r \rangle$	6.140	
$\langle r^2 \rangle$	17.339	

WAVEFUNCTION

n	η	c(1s)	c(2s)
1	3.684801	0.997586	-0.204439
2	0.956031	0.012388	1.018244

* These values were computed using the above wavefunction in computing the $\langle H^2 \rangle$ matrices.

TABLE IV-10

Be $1s^2\ 2s^2$ ^1S Δ-minimization

EXP.ENERGY	-14.66785
CALC.ENERGY	-14.378674
$\langle H^2 \rangle$	207.922944
EPSILON	- 0.289172
DELTA	1.176655
DELTA-TILDE	1.260275
$\langle 1/r \rangle$	9.165
$\langle r \rangle$	4.789
$\langle r^2 \rangle$	9.971

WAVEFUNCTION

n	η	c(1s)	c(2s)
1	3.843778	0.968926	-0.386187
2	1.290642	0.094243	1.038741

TABLE IV-11

Be $1s^2\ 2s^2$ ^1S $\tilde{\Delta}$-minimization

EXP.ENERGY	-14.66785
CALC.ENERGY	-14.444846
$\langle H^2 \rangle$	209.814687
EPSILON	-0.223000
DELTA	1.161095
DELTA-TILDE	1.210824
$\langle 1/r \rangle$	9.072
$\langle r \rangle$	5.880
$\langle r^2 \rangle$	11.426

WAVEFUNCTION

n	η	c(1s)	c(2s)
1	3.865673	0.975836	-0.338806
2	1.195039	0.083379	1.029698

TABLE IV-12

Be $1s^2\ 2s^2$ ^1S $\epsilon^2/\tilde{\Delta}$ -minimization

EXP.ENERGY	-14.66785
CALC.ENERGY	-14.506959
$\langle H^2 \rangle$	216.338208
EPSILON	0.160887
DELTA	5.886334
DELTA-TILDE	5.912219
$\langle 1/r \rangle$	8.144
$\langle r \rangle$	6.234
$\langle r^2 \rangle$	17.779

WAVEFUNCTION

n	η	c(1s)	c(2s)
1	3.557376	1.000355	-0.198810
2	0.945765	-0.001181	1.019017

TABLE IV-13

Be $1s^2\ 2s^2\ \ {}^1S$ ϵ^2/Δ -minimization

EXP.ENERGY	-14.66785
CALC.ENERGY	-14.506958
$\langle H^2 \rangle$	216.338298
EPSILON	- 0.160888
DELTA	5.886455
DELTA-TILDE	5.912340
$\langle 1/r \rangle$	8.143
$\langle r \rangle$	6.234
$\langle r^2 \rangle$	17.779

WAVEFUNCTION

n	η	c(1s)	c(2s)
1	3.557360	1.000356	-0.198807
2	0.945767	-0.001819	1.019918

The starting wavefunction and the starting orbital exponents were taken from the work of Huzinaga (19). In order to have one more test of the correctness of the program used, the $\langle H \rangle$ minimization was repeated in full with optimization of the orbital exponents and the results obtained agreed to 6 - 8 figures with Huzinaga's results. The wavefunction obtained in the $\langle H \rangle$ minimization was then used to compute the expectation value $\langle H^2 \rangle$ to obtain a result for Δ and $\tilde{\Delta}$ which could be compared with the other variational methods.

The following general trends hold for the various quantities computed:

$$E(\Delta) \cong E(\tilde{\Delta}) > E(\epsilon^2/\Delta) = E(\epsilon^2/\tilde{\Delta}) > E(\langle H \rangle)$$

$$\Delta(\epsilon^2/\Delta) \cong \Delta(\epsilon^2/\tilde{\Delta}) > \Delta(\langle H \rangle) > \Delta(\Delta) \cong \Delta(\tilde{\Delta})$$

Furthermore the wavefunctions obtained by ϵ^2/Δ and $\epsilon^2/\tilde{\Delta}$ minimization resemble closely those obtained from $\langle H \rangle$ minimization, whereas the wavefunctions for $\tilde{\Delta}$ and Δ resemble each other but are clearly distinct from those obtained by the other methods. The deterioration of the energy for delta and delta-tilde minimizations was quite striking.

IV. 7 Overlap between SCF- and CI-wavefunctions.

In order to obtain a more thorough understanding of the methods employed the overlap $a = \langle\phi|\Psi\rangle$ of the computed SCF-wavefunctions with various "correct" wavefunctions was calculated. Table (IV-14) displays these results. The so-called "correct" wavefunctions were various configuration interaction wavefunctions (28,29,30) or in the case of hydrogen the correct ground state wavefunction. The computation of the Be overlap provided a surprising result, the overlap obtained from $\epsilon^2/\tilde{\Delta}$ minimization was less than that for the $\langle H\rangle$ minimization. To test if this result was due to a bad approximate wavefunction or to electron correlation, the overlaps of various approximate wavefunctions, ϕ, of Hydrogen, with the exact wavefunction were computed. These approximate wavefunctions were chosen to be a linear combination of two STO's i.e. $\phi = c_1\chi_1 + c_2\chi_2$ .The orbital exponents were chosen in the range from 0.1 to 20.0. The coefficients c_1 and c_2 were chosen in such a way that the quantity under consideration was minimized. One result of the computation (orbital exponents 0.4 and 2.0) is displayed in table IV-14. For hydrogen $\epsilon^2/\tilde{\Delta}$ variation led to a maximization of the overlap in all cases.

TABLE IV-14

Overlap between SCF- and CI-wavefunctions

ATOM	CONFIGURATION	STATE	METHOD*	OVERLAP
H	1s	^2S	1	0.884437
			4	0.927895
			3	0.772312
He	$1s^2$	^1S	1	0.996202
			4	0.995271
	1s2s	^3S	1	0.903707
			4	0.914973
Be	$1s^2 2s^2$	^1S	1	0.956581
			2	0.861201
(2 basis functions)			3	0.901292
			4	0.956198
			5	0.956197
(4 basis functions)			1	0.957503
			4	0.957108
			5	0.957116

*see Table IV-1 for an explanation of the methods

[illegible]

IV. 8 Summary and conclusions.

In this thesis variational principles have been coded for an electronic computer so that they could be applied to atomic states and their usefulness be assessed.

The thesis has clearly demonstrated that the minimization of Δ or $\tilde{\Delta}$ does not lead to wavefunctions that are useful in obtaining physical properties of the state under consideration. From the calculated overlaps it can be concluded that both methods yield wavefunctions that occupy different parts of configuration space than the true wavefunction. This finding is similar to the results obtained by Messmer and Birss (3,4). They calculated wavefunctions using the Temple-Kato bound (23,24) and the Lowdin bounds (8). These authors found that the bound formulations add contributions to the trial wavefunction which do not help in the description of the wavefunction associated with the state being considered. On the other hand the computaion of Δ or $\tilde{\Delta}$ using the SCF-method leads0 to a very unsatisfactory bound. According to Weinstein (6) the energy of the true wavefunction lies somewhere between $E + \sqrt{\Delta} \geqslant W \geqslant E - \sqrt{\Delta}$. In no system but He has it been possible to obtain a value of delta smaller than about 1. This bound is really very unsatisfactory if one tries to associate a wavefunction with an excited state. For example a typical energy difference between states for

smaller atoms is about 0.1 a.u.. Therefore, to assign a definite excited state to a certain wavefunction, delta should be ≤ 0.01. It should be pointed out that the size of the bound is not necessarily connected with the quality of the wavefunction. This can be seen for the Be ground state calculation, where the $\epsilon^2/\tilde{\Delta}$-minimization yields a far better wavefunction than the $\tilde{\Delta}$-minimization, even though the bounds in the former method are much worse than the bounds in the delta minimization. The better quality shows itself in the better energy, better overlap and $\langle 1/r \rangle$, $\langle r \rangle$ and $\langle r^2 \rangle$ values.

The minimization of ϵ^2/Δ or $\epsilon^2/\tilde{\Delta}$ yields good wavefunctions for ground and excited states. The main problem consists of assigning definite configurations to each wavefunction and each minimum. In the $\epsilon^2/\tilde{\Delta}$ method one minimizes the overlap of the "correction" wavefunction ϕ_χ:

$$a_\chi^2 = \langle \phi_\chi | \phi_\chi \rangle$$

since the relation

$$\epsilon^2/\tilde{\Delta} \leq a_\chi^2 \leq \tilde{\Delta}/(W_k - W_k^!)^2$$

holds. In all cases the value of $\epsilon^2/\tilde{\Delta}$ could be reduced to about 0.01 or less., whereas the $\tilde{\Delta}/(W_k - W_k^!)^2$ value was of the order 100 to 200. One is therefore hard put to assign a certain configuration to the wavefunction obtained. It is strongly believed that this inability to reduce the upper bound of a_χ^2 reasonably is connected with the orbital

expansion SCF approach. The best wavefunction one can expect from this approach is a Hartree-Fock type wavefunction with all its shortcomings with respect to the correlation of the electrons. This aspect of the variational methods is further clarified by analyzing the overlaps for the wavefunctions obtained by the SCF-methods and CI-methods.

The overlaps obtained by minimizing $\langle H \rangle$ and $\epsilon^2/\tilde{\Delta}$ (or ϵ^2/Δ) show a peculiar behaviour. The theory states that one should obtain maximum overlap of the trial wavefunction with the true wavefunction by minimizing $\epsilon^2/\tilde{\Delta}$ and this result is indeed obtained for H 1s ^1S and He 1s2s ^3S, but for the other states the wavefunction obtained by $\langle H \rangle$-minimization leads invariably to a larger overlap. All these results clearly indicate the sensitivity of the $\epsilon^2/\tilde{\Delta}$ methods to correlation between electrons. For the states where the method leads to the expected maximization of the overlap the correlation between electrons is nonexistent or small. For the states where correlation is important and large (e.g. two electrons occupying one space orbital) the overlap of the wavefunction obtained by $\epsilon^2/\tilde{\Delta}$ minimization is smaller than the overlap of the wavefunction obtained by $\langle H \rangle$-minimization. Furthermore the size of delta is proportional to the importance of correlation. This is most clearly demonstrated by the two

states He $1s^2$ ^1S and Li^+ $1s^2$ ^1S. (These results have not been tabulated). These two atomic systems are very similar in many aspects. The larger charge of the Li-nucleus exerts a larger attractive force upon the electrons thereby diminishing the average distance from the nucleus as compared with He. The shorter distance of the electrons from the nucleus enhances the correlation between them. This enhancement is reflected by an increase of from 0.5 (He $1s^2$ ^1S) to 1.4 (Li^+ $1s^2$ ^1S). An answer to the question of how critical the correlation effect between electrons is for the $\epsilon^2/\tilde{\Delta}$ variational method could be obtained by employing various configuration interaction wave functions to the $\epsilon^2/\tilde{\Delta}$ variation. One should then be able to observe the amount of correlation which has to be taken into account before a wavefunction obtained by $\epsilon^2/\tilde{\Delta}$ minimization shows a greater overlap with a "true" wavefunction than one obtained by minimizing $\langle H \rangle$.

BIBLIOGRAPHY

1. R. P. Messmer, Theoret. Chim. Acta. (Berl.) 14, 319 (1969).

2. J. H. Choi, C. F. Lebeda and R. P. Messmer, Chem. Phys. Lett. 5, 503 (1970).

3. R. P. Messmer and F. W. Birss, Theoret. Chim. Acta. (Berl.) 14, 192 (1969).

4. R. P. Messmer and F. W. Birss, Theoret. Chim. Acta. (Berl.) 14, 203 (1969).

6. D. H. Weinstein, Proc. Natl. Acad. Sci. (U.S.) 20, 529 (1934).

7. J. K. L. MacDonald, Phys. Rev. 46, 828 (1934).

8. P. O. Lowdin, Phys. Rev. 139, A 357 (1965).

9. S. Fraga and F. W. Birss, J. Chem. Phys. 40, 3207 (1964)

10. R. P. Messmer, Ph.D. Thesis,University of Alberta (1967)

11. H. M. James and A. S. Coolidge, Phys. Rev. 51, 860 (1937).

12. F. W. Birss and U. Liebe, unpublished results.

13. J. Goodisman, J. Chem. Phys. 45, 3659 (1966).

14. C. C. J. Roothaan, Rev. Mod. Phys. 23, 69 (1951) and 32 179 (1960).

15. M. Tinkham, Group Theory and Quantum Mechanics, McGraw Hill, 1964 pp 173.

16. H.F. Schaefer and F. E. Harris, J. of Computational

Physics, 3, 217 (1968).

17. F. W. Birss and S. Fraga, J. Chem. Phys.38, 2552 (1963) and 40, 3203 (1964).

18. S. Huzinaga, J. Chem. Phys. 51, 3971 (1969).

19. S. Huzinaga, Approximate Atomic Functions, Department of Chemistry, University of Alberta, 1971.

20. J. Hinze and C. C. J. Roothaan, Suppl. Progr. Theor. Phys. 40, 37 (1967).

21. F. L. Pilar, Elementary Quantum Chemistry, McGraw Hill, (1968).

22. M. Hamermesh, Group Theory and Its Applications to Physical Problems, Addison-Wesley (1964), Chapters 1 - 2

23. G. Temple, Proc. Roy. Soc. (London), A 119, 276 (1928).

24. T. Kato, J. Phys. Soc. Japan 4, 334 (1949).

25. R. P. Madden and K. Codling, Astrophys. J. 141, 364 (1965).

26. C. C. J. Roothaan and P. S. Bagus in Methods in Computational Physics Vol. 2, Acad. Press 1963, pp47.

27. J. C. Slater Phys.Rev. 36, 57 (1930)

28. R. E. Watson Phys. Rev. 119, 170 (1960)

29. R. K. Nesbet and R. E. Watson, Phys.Rev. 110,1073 (1958)

30. H. L. Davis J. Chem. Phys. 37, 1508 (1962)

31. E.R. Davidson J. Chem. Phys. 42, 4199 (1965)

32. J.K.L. MacDonald, Phys. Rev. 43,830 (1933)

APPENDIX I.

THE EVALUATION OF MATRIX ELEMENTS OF 3- and 4- ELECTRON OPERATORS BETWEEN DETERMINANTAL WAVEFUNCTIONS.

Let Q be a product of orthonormal spinorbitals (15)

$$Q = \prod_i \phi_i$$

Then the determinantal wavefunction can be written as

$$\det Q = AQ$$

where A, the antisymmetrizer is given by

$$A = (N!)^{-1/2} \sum_p (-1)^p P$$

and the P are the operators which form N! different permutations of the subscripts of the ϕ_i.

The expectation value of a quantum mechanical operator that commutes with the antisymmetrizer can be shown (22) to be

$$\langle O \rangle = \langle AQ|O|AQ \rangle = \sum_p (-1)^p \langle Q|O|PQ \rangle \qquad \text{(I-1,1)}$$

Let O_m be an operator that operates on m electrons. Then only permutations belonging to a symmetry group S_n such that n m will contribute to the expectation value of O_m.

EXAMPLE:

Let $\phi(i)\phi(j)\phi(k)\phi(l)$ be a product of orthonormal spinorbitals and let the determinantal wavefunction be given by

$$D = A|\phi(i)\phi(j)\phi(k)\phi(l)\rangle$$

Then the expectation value of an operator O^{123} is given by:

$$\langle O^{123}\rangle = \sum (-1)^{P} \langle\phi(i;1)\phi(j;2)\phi(k;3)\phi(l;4)|O^{123}| P\,\phi(i;1)\phi(j;2)\phi(k;3)\phi(l;4)\rangle$$

Let P = (1 2 3 4) [for an explanation of this notation of permutations see (23)] then this particular term is given by

$$\langle\phi(i;1)\phi(j;2)\phi(k;3)\phi(l;4)|O^{123}|\phi(l;1)\phi(i;2)\phi(j;3)\phi(k;4)\rangle$$
$$= \langle\phi(i;1)\phi(j;2)\phi(k;3)|O^{123}|\phi(l;1)\phi(i;2)\phi(j;3)\rangle \langle\phi(l:4)|\phi(k;4)\rangle = 0$$

since

$$\langle\phi(l)|\phi(k)\rangle = \delta lk$$

Therefore the expectation value of the 3- and 4-electron operators can be written as

$$O^{3} = \sum i<j<k \; O^{ijk} \qquad \text{and} \qquad O^{4} = \sum i<j<k<l \; O^{ijkl}$$

are expressed as

$$\langle O^3 \rangle = \sum_{i<j<k} \sum_{P} (-1)^P \langle \phi(i;1)\phi(j;2)\phi(k;3) | O'^{23} | P\,\phi(i;1)\phi(j;2)\phi(k;3) \rangle$$

and

$$\langle O^4 \rangle = \sum_{i<j<k<l} \sum_{P} (-1)^P \langle \phi(i;1)\phi(j;2)\phi(k;3)\phi(l;4) | O'^{234} | P\,\phi(i;1)\phi(j;2)\phi(k;3)\phi(l;4) \rangle$$

Since there are 6 elements of S_3 and 24 elements of S_4, it does not seem fruitful to carry the expansion further as in the case of the 2-electron operators and to give each permuted integral a special name and a special symbol (such as J_{ij} and K_{ij} in the 2-electron case).

APPENDIX II.

Attention should be drawn to a property of the integrals when P is either a 3-cycle or a 4-cycle permutation, when these integrals are subjected to a variation. An example will better clarify this particularity than a general discussion.

Consider the particular term

$$\sum_{i<j<k} \langle \phi(i;1)\phi(j;2)\phi(k;3)|O^{123}|\phi(k;1)\phi(i;2)\phi(j;3)\rangle$$

where P = (1 2 3). Variation of the particular orbital ϕ_i yields:

$$\sum_{i<j<k} \langle \delta\phi(i;1)\{\phi(j;2)\phi(k;3)|O^{123}|\phi(k;1)\phi(j;3)\}\phi(i;2)\rangle$$
$$+\text{complex conjugate}$$

In the SCF-formalism the expression

$$\sum_{j>i,\,k>j}\{\phi(j;2)\phi(k;3)|O^{123}|\phi(k;1)\phi(j;3)\}$$

contributes towards the F-matrix connected with the orbital $\phi(i)$.

But this operator is not necessarily Hermitian, i.e.

$$\{\phi(j;2)\phi(k;3)|O^{123}|\phi(k;1)\phi(j;3)\}^{+} \neq \{\phi(j;2)\phi(k;3)|O^{123}|\phi(k;1)\phi(j;3)\}$$

in general.

This non-hermiticity has to be accounted for by a suitable averaging process.

EXAMPLE:

Let $\phi(i) = 1s$ $\phi(j) = 2s$ $\phi(k) = 3d^{\circ}$. Let each of these orbitals be expanded into a suitable set of basis functions, e.g.

$$1s = \sum_i c(1s;i)\,\chi(s;i)$$
$$2s = \sum_i c(2s;i)\,\chi(s;i)$$
$$3d^{\circ} = \sum_i c(3d;i)\,\chi(d^{\circ};i)$$

then

$$\{\phi(j;2)\phi(k;3)|O^{123}|\phi(k;1)\phi(j;3)\}^{mn} =$$

$$\sum_{i,j,k,l}[c(s;i)c(s;j)c(d;k)c(d;l) \langle\chi(s;m)\chi(s;i)\chi(d^{\circ};k)|O^{123}|\chi(d^{\circ};l)\chi(s;n)\chi(s;j)\rangle]$$

whereas

$$\{\phi(j;2)\phi(k;3)|O^{123}|\phi(k;1)\phi(j;3)\}^{nm} =$$
$$\sum_{i,j,k,l} [c(s;i)c(s;j)c(d;k)c(d;l)$$
$$\langle\chi(s;m)\chi(s;i)\chi(d^{\circ};k)|O^{123}|\chi(d^{\circ};l)\chi(s;n)\chi(s;j)\rangle]$$

The summations over the expansion coefficients are the same.

The operator O^{123} is given by

$$O^{123} = 2*[h^1(r^{23})^{-1} + h^2(r^{13})^{-1} + h^3(r^{12})^{-1}] + (r^{12}r^{13})^{-1}$$
$$(r^{13}r^{12})^{-1} + (r^{12}r^{23})^{-1} + (r^{23}r^{12})^{-1} + (r^{13}r^{23})^{-1} + (r^{23}r^{13})^{-1}$$

This, applied to the particular case above yields for the first expression

$$\sum_{i,j,k,l} \{c(s;i)c(s;j)c(d;k)c(d;l)$$
$$*[\langle\chi(s;m)|h|\chi(d^{\circ};l)\rangle\langle\chi(s;i)\chi(d^{\circ};k)|1/r^{23}|\chi(s;n)\chi(s;j)\rangle$$
$$+\langle\chi(s;i)|h|\chi(s;m)\rangle\langle\chi(s;n)\chi(d^{\circ};k)|1/r^{13}|\chi(d^{\circ};l)\chi(s;j)\rangle$$
$$+\langle\chi(d^{\circ};k)|h|\chi(s;j)\rangle\langle\chi(s;n)\chi(s;i)|1/r^{12}|\chi(d^{\circ};l)\chi(s;m)\rangle$$
$$+\langle\chi(s;i)\chi(d^{\circ};k)\{\chi(s;n)|\chi(d^{\circ};l)\}\chi(s;m)\chi(s;j)\rangle$$
$$+\langle\chi(d^{\circ};k)\chi(s;i)\{\chi(s;n)|\chi(d^{\circ};l)\}\chi(s;m)\chi(s;j)\rangle$$
$$+\langle\chi(s;n)\chi(d^{\circ};k)\{\chi(s;i)|\chi(s;j)\}\chi(d^{\circ};l)\chi(s;j)\rangle$$
$$+\langle\chi(d^{\circ};k)\chi(s;n)\{\chi(s;i)|\chi(s;m)\}\chi(s;j)\chi(d^{\circ};l)\rangle$$
$$+\langle\chi(s;n)\chi(s;i)\{\chi(d^{\circ};k)|\chi(s;j)\}\chi(d^{\circ};l)\chi(s;m)\rangle$$
$$+\langle\chi(s;i)\chi(s;n)\{\chi(d^{\circ};k)|\chi(s;j)\}\chi(s;m)\chi(d^{\circ};l)\rangle]\}$$

and for the second expression

$$\sum_{i,j,k,l} \{c(s;i)c(s;j)c(d;k)c(d;l)$$
$$*[\langle\chi(s;m)|h|\chi(d^\circ;l)\rangle\langle\chi(s;i)\chi(d^\circ;k)|1/r^{23}|\chi(s;n)\chi(s;j)\rangle$$
$$+\langle\chi(s;i)|h|\chi(s;n)\rangle\langle\chi(s;m)\chi(d^\circ;i)|1/r^{13}|\chi(d^\circ;l)\chi(s;j)\rangle$$
$$+\langle\chi(d^\circ;k)|h|\chi(s;j)\rangle\langle\chi(s;m)\chi(s;i)|1/r^{12}|\chi(d^\circ;l)\chi(s;n)\rangle$$
$$+\langle\chi(s;i)\chi(d^\circ;k)\{\chi(s;m)|\chi(d^\circ;l)\}\chi(s;n)\chi(s;j)\rangle$$
$$+\langle\chi(d^\circ;k)\chi(s;i)\{\chi(s;m)|\chi(d^\circ;l)\}\chi(s;j)\chi(s;n)\rangle$$
$$+\langle\chi(s;j)\chi(d^\circ;k)\{\chi(s;i)|\chi(s;n)\}\chi(d^\circ;l)\chi(s;j)\rangle$$
$$+\langle\chi(d^\circ;k)\chi(s;j)\{\chi(s;i)|\chi(s;n)\}\chi(s;j)\chi(d^\circ;l)\rangle$$
$$+\langle\chi(s;m)\chi(s;i)\{\chi(d^\circ;k)|\chi(s;j)\}\chi(d^\circ;l)\chi(s;n)\rangle$$
$$+\langle\chi(s;i)\chi(s;m)\{\chi(d^\circ;k)|\chi(s;j)\}\chi(s;m)\chi(d^\circ;l)\rangle]\}$$

where

$$\langle\chi(s;i)\chi(d^\circ;k)\{\chi(s;m)|\chi(d^\circ;l)\}\chi(s;n)\chi(s;j)\rangle$$

symbolizes the integral

$$\int\{\chi(s;i;r^1)\chi(s;n;r^1)\chi(d^\circ;k;r^2)\chi(s;j;r^2)\chi(s;m;r^3)\chi(d^\circ;l;r^3)$$
$$/(r^{13}r^{23})\}\, dr^1dr^2dr^3$$

It can be seen that the two expression are different.

APPENDIX III.

THE CONSTRUCTION OF L-S EIGENFUNCTIONS.

This appendix describes how one can obtain a wavefunction $\phi(2s+1:l)$ belonging to a certain configuration expressed as a sum of slators, such that this wavefunction is an eigenfunction of the operators L^2, L_z, S^2, S_z

$$L^2\phi(2s+1:l) = l(l+1)\phi(2s+1:l)$$
$$L_z\phi(2s+1:l) = ml\phi(2s+1:l)$$
$$S^2\phi(2s+1:l) = s(s+1)\phi(2s+1:l)$$
$$S_z\phi(2s+1:l) = ms\phi(2s+1:l)$$

This discussion is based upon a suggestion by Schaeffer and Harris (17), but it is written with the aim to particularize and clarify some points important to the present work.

To aid the reader not familiar with the concepts, a specific example (p^2 ^1S) is given at the end of this appendix.

As is well known (22) the operation of the L and S operators can be expressed as:

$$S_zD = 1/2(n_\alpha - n_\beta)D = M_SD$$

$$L_z D = \sum_{i=1,N} ml^i D = M_L D$$

$$S^2 D = \{ \sum P_{\alpha\beta} + (1/4)[(n_\alpha - n_\beta)^2 + 2n_\alpha + 2n_\beta] \} D$$
$$L^2 D = \{ L^- L^+ + L_z(L_z+1) \} D$$

with

$$L^\pm = \sum_{i=1,N} \{ L^\pm (i) \}$$

Each slator is automatically an eigenfunction of L_z and S_z.

To obtain a linear combination of slators which is an eigenfunction of L^2 and S^2, one collects all slators belonging to the configuration in question which have a L_z eigenvalue of ml=l and a S_z eigenvalue of ms=s into a vector

$$d = (D^1\ D^2\ D^3\ \ldots\ldots\ldots D^n).$$

One then forms the matrix

$$[\underset{\sim}{LS}] = d^+(L^2 + kS^2)d$$

and diagonalizes it. A proper choice of k gives a well spaced eigenvalue spectrum. By including only slators with ml=l and ms=s one has assured that the lowest eigenvalue a[1]

of the matrix $[\underset{\sim}{LS}]$ is

$$a^{1} \geqslant ml(ml+1)+k*ms(ms+1)$$

If the equality holds, then the eigenvector associated with this eigenvalue is the required linear combination. If for more than one eigenvector this equality holds, one has a case of degeneracy. If the equality does not hold then no linear combination of slators for this configuration possesses the required symmetry.

From the linear combinations with ml=l and ms=s one obtains all linear combinations with l≤ml≤-l and s≤ms≤-s by repeatedly applying the operators L^- and S^-.

EXAMPLE ($p^2\ ^1S$)

The distinct slators for this configuration are:

$$D^1 = |p^{+1}(1)\beta(1)\ p^{-1}(2)\alpha(2)| = |\overline{p^{+1}}\ p^{-1}|$$

$$D^2 = |p^{+1}(1)\alpha(1)\ p^{-1}(2)\beta(2)| = |p^{+1}\overline{p^{-1}}|$$

$$D^3 = |p^{\circ}(1)\alpha(1)\ p^{\circ}(2)\beta(2)| = |p^{\circ}\overline{p^{\circ}}|$$

where the two vertical bars designate a determinant i.e.

$$|p^{-1}\overline{p^{+1}}| = p^{-1}(1)\alpha(1)*p^{+1}(2)\beta(2) - p^{-1}(2)\alpha(2)*p^{+1}(1)\beta(1)$$

Operation with the operator L^2+S^2 gives the result:

$$S^2D^1 = S^2|\overline{p^{+1}}\ p^{-1}| = |p^{+1}\overline{p^{-1}}|+|\overline{p^{+1}}\ p^{-1}| = D^2+D^1$$

$$L^2D^1 = L^2|\overline{p^{+1}}\ p^{-1}| = [L^-L^+ + L_z(L_z+1)]|\overline{p^{+1}}\ p^{-1}|$$

$$= L^-L^+|\overline{p^{+1}}\ p^{-1}|+0*|\overline{p^{+1}}\ p^{-1}|$$

$$= [L^-(1)+L^-(2)]*[L^+(1)+L^+(2)]|\overline{p^{+1}}\ p^{-1}|$$

$$= [L^-(1)+L^-(2)]|\overline{p^{+1}}p^{\circ}|$$

$$= |\overline{p^{\circ}}p^{\circ}|+|\overline{p^{+1}}\ p^{-1}| = -D^3+D^1$$

Therefore

$$(L^2+S^2)D^1 = 2D^1+D^2-D^3$$

Similarly

$$(L^2+S^2)D^2 = 2D^2+D^1+D^3$$

$$(L^2+S^2)D^3 = 2D^3-D^1+D^2$$

Since the relationship $\langle D^I | D^J \rangle = \delta_{IJ}$ holds, the matrix $[\underset{\sim}{LS}]$ is easily seen to be:

$$[\underset{\sim}{LS}] = \begin{pmatrix} 2 & 1 & -1 \\ 1 & 2 & 1 \\ -1 & 1 & 2 \end{pmatrix}$$

The matrix of eigenvectors is given by

$$\begin{pmatrix} 1/\sqrt{2} & -1/\sqrt{3} & -1/\sqrt{6} \\ 1/\sqrt{2} & 1/\sqrt{3} & -1/\sqrt{6} \\ 0 & -1/\sqrt{3} & 2/\sqrt{6} \end{pmatrix}$$

with the eigenvalues 3,0,3. The normalized linear combination of slators for the configuration p^2 and the state ^1S is therefore :

$$\phi(^1S) = (1/\sqrt{3})(-D^1+D^2-D^3)$$

APPPENDIX IV.

EXTENSION OF THE HINZE-ROOTHAAN FORMALISM TO THE CASE OF 3- AND 4-ELECTRON OPERATORS.

The F-matrix connected with each orbital is given by eqn. (3-17). Following the derivation of Hinze and Roothaan one computes a correction matrix to the F(n,l)-matrix by considering all first order changes in the F(n,l)-matrix which are caused by a change of the vectors c(n',l), where n' runs over all orbitals c(n',l) that belong to the symmetry l.

Using eqn. (3-17) one writes the correction matrix as (we assume that the orbitals are in real form):

$$\delta F^{nl}_{pq} =$$

$$2\times\sum_{j=1}^{I^2} B^j \sum_{\alpha\neq\beta\;1}^{2} \sum_{r,s} \delta c_r^{n_j^\alpha l}\, c_s^{n_j^\alpha l}\, \delta_{n_j^\beta n}\, \delta_{l_j^\beta l}\, \delta_{l_j^\alpha l} \cdot J^{P_\alpha\,(pq;rs)}_{\varrho_j^\alpha\,\varrho_j^\beta}$$

$$+\, 2\times\sum_{k=1}^{I^3} C^k \sum_{\substack{\alpha\neq\beta\\ \alpha\neq\gamma\\ \beta\neq\gamma}\;1}^{3} \sum_{\substack{r,s\\ t,u}} \delta c_r^{n_k^\alpha l_k^\alpha}\, c_s^{n_k^\alpha l_k^\alpha}\, c_t^{n_k^\beta l_k^\beta}\, c_u^{n_k^\beta l_k^\beta}\, \delta_{n_k^\gamma n}\, \delta_{l_k^\gamma l}\, \delta_{l_k^\alpha l}$$

$$K^{P_\alpha\,(pq;rs;tu)}_{\varrho_k^\alpha\,\varrho_k^\beta\,\varrho_k^\gamma}$$

$$+\, 2\times\sum_{l=1}^{I^4} D^l \sum_{\substack{\alpha\neq\beta\\ \alpha\neq\gamma\;\; \alpha\neq\delta\\ \beta=\gamma\;\; \beta\neq\delta\\ \gamma\neq\delta}\;1}^{4} \sum_{\substack{r,s\\ t,u\\ v,w}} \delta c_r^{n_\ell^\alpha l_\ell^\alpha}\, c_s^{n_\ell^\alpha l_\ell^\alpha}\, c_t^{n_\ell^\beta l_\ell^\beta}\, c_u^{n_\ell^\beta l_\ell^\beta}\, c_v^{n_\ell^\gamma l_\ell^\gamma}\, c_w^{n_\ell^\gamma l_\ell^\gamma}\, \delta_{n_\ell^\delta n}\, \delta_{l_\ell^\delta l}\, \delta_{l_\ell^\alpha l}$$

$$L^{P_\alpha\,(pq;rs;tu;vw)}_{\varrho_\ell^\alpha\,\varrho_\ell^\beta\,\varrho_\ell^\gamma\,\varrho_\ell^\delta}$$

Equation (52) of ref. (21)

$$\delta \underset{\sim}{F}^{nl}\, \underset{\sim}{c}^{nl} = 2\times\sum_{n'} \underset{\sim}{\mathcal{L}}^{l}_{nn'}\, \delta c^{n'l}$$

still holds, but the definition of the L-matrix

has changed to:

$$\mathcal{L}^{l}_{nn';pq} =$$

$$2 \times \sum_{j=1}^{I^2} B^j \sum_{\alpha\neq\beta}^{2}{}_{1} \sum_{r,s} \delta\, C_r^{n_j^\alpha l} C_s^{n_j^\alpha l}\, \delta_{n_j^\beta n}\, \delta_{l_j^\beta l}\, \delta_{l_j^\alpha l} \cdot J^{P_\alpha (pr;qs)}_{\rho_j^\alpha \rho_j^\beta}$$

$$+ 2 \times \sum_{k=1}^{I^3} C^k \sum_{\substack{\alpha\neq\beta \\ \alpha\neq\gamma \\ \beta\neq\gamma}}^{3}{}_{1} \sum_{\substack{r,s \\ t,u}} \delta\, C_r^{n_k^\alpha l_k^\alpha} C_s^{n_k^\alpha l_k^\alpha} C_t^{n_k^\beta l_k^\beta} C_u^{n_k^\beta l_k^\beta}\, \delta_{n_k^\gamma n}\, \delta_{l_k^\gamma l}\, \delta_{l_k^\alpha l}$$

$$K^{P_\alpha (pr;qs;tu)}_{\rho_k^\alpha \rho_k^\beta \rho_k^\gamma}$$

$$+ 2 \times \sum_{l=1}^{I^4} D^l \sum_{\substack{\alpha\neq\beta \\ \alpha\neq\gamma\ \alpha\neq\delta \\ \beta\neq\gamma\ \beta\neq\delta \\ \gamma\neq\delta}}^{4}{}_{1} \sum_{\substack{r,s \\ t,u \\ v,w}} \delta\, C_r^{n_l^\alpha l_l^\alpha} C_s^{n_l^\alpha l_l^\alpha} C_t^{n_l^\beta l_l^\beta} C_u^{n_l^\beta l_l^\beta} C_v^{n_l^\gamma l_l^\gamma} C_w^{n_l^\gamma l_l^\gamma}\, \delta_{n_l^\delta n}\, \delta_{l_l^\delta l}\, \delta_{l_l^\alpha l}$$

$$L^{P_\alpha (pr;qs;tu;vw)}_{\rho_l^\alpha \rho_l^\beta \rho_l^\gamma \rho_l^\delta}$$

With this definition of the L-matrix the formalism of (21) can proceed unchanged to its end.

APPENDIX V.

EXAMPLE: Be $1s^2 2s\ 3d\ ^1D$

Equation (3-1) for this configuration is given by

$$\phi(^1D) = (\sqrt{2})^{-1}\ [|1s\bar{1s}2s\overline{3d^{+2}}| - |1s\bar{1s}\bar{2s}3d^{+2}|]$$
$$= (\sqrt{2})^{-1}\ (D^1 - D^2)$$

The expectation value of the operator 0 eqn (3-6) is:

$$\langle 0\rangle = (1/2)\langle D^1|0|D^1\rangle - 2*(1/2)\langle D^1|0|D^2\rangle + (1/2)\langle D^2|0|D^2\rangle$$

This, rewritten in the form (3-7), yields:

$$\begin{aligned}\langle 0\rangle = \quad & (1/2)\ \langle 1s\bar{1s}2s\overline{3d^{+2}}|0|\textstyle\sum(-1)^p\, P\ 1s\bar{1s}2s\overline{3d^{+2}}\rangle \\ - \quad & \langle 1s\bar{1s}2s\overline{3d^{+2}}|0|\textstyle\sum(-1)^p\, P\ 1s\bar{1s}\bar{2s}3d^{+2}\rangle \\ +(1/2) \quad & \langle 1s\bar{1s}\bar{2s}3d^{+2}|0|\textstyle\sum(-1)^p\, P\ 1s\bar{1s}\bar{2s}3d^{+2}\rangle\end{aligned}$$

Inserting the explicit form of the operator 0 the form (3-9) is obtained.

$$\begin{aligned}\langle 0\rangle = (1/2)\ & \langle 1s|0^1|1s\rangle + \langle\bar{1s}|0^1|\bar{1s}\rangle + \langle 2s|0^1|2s\rangle \\ & + \langle\overline{3d^{+2}}|0^1|\overline{3d^{+2}}\rangle \\ & + \langle 1s\bar{1s}|0^2|1s\bar{1s}\rangle - \langle 1s\bar{1s}|0^2|\bar{1s}1s\rangle + \langle 1s2s|0^2|1s2s\rangle \\ & - \langle 1s2s|0^2|2s1s\rangle + \langle 1s\overline{3d^{+2}}|0\ |1s\overline{3d^{+2}}\rangle - \langle 1s\overline{3d^{+2}}|0^2|\overline{3d^{+2}}1s\rangle\end{aligned}$$

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

$$+\langle\overline{1s}2s|O^2|\overline{1s}2s\rangle+\langle\overline{1s}\,\overline{3d^{+2}}|O^2|\overline{1s}\,\overline{3d^{+2}}\rangle-\langle\overline{1s}\,\overline{3d^{+2}}|O^2|\overline{3d^{+2}}\,\overline{1s}\rangle$$

$$-\langle\overline{1s}2s|O^2|2s\overline{1s}\rangle+\langle 2s\overline{3d^{+2}}|O^2|2s\overline{3d^{+2}}\rangle-\langle 2s\overline{3d^{+2}}|O^2|\overline{3d^{+2}}2s\rangle$$

$$+\langle 1s\overline{1s}2s|O^3|\sum(-1)^p P\ 1s\overline{1s}2s\rangle$$

$$+\langle 1s\overline{1s}\,\overline{3d^{+2}}|O^3|\sum(-1)^p P\ 1s\overline{1s}\,\overline{3d^{+2}}\rangle$$

$$+\langle 1s2s\overline{3d^{+2}}|O^3|\sum(-1)^p P\ 1s2s\overline{3d^{+2}}\rangle$$

$$+\langle\overline{1s}2s\overline{3d^{+2}}|O^3|\sum(-1)^p P\ \overline{1s}2s\overline{3d^{+2}}\rangle$$

$$+\langle 1s\overline{1s}2s\overline{3d^{+2}}|O^4|\sum(-1)^p P\ 1s\overline{1s}2s\overline{3d^{+2}}\rangle$$

+ the terms arising from $\langle D^1|O|D^2\rangle$ and $\langle D^2|O|D^2\rangle$

Collecting all the identical integrals and summing their coefficients, yields the eqn (3-10):

$$\begin{aligned}
\langle O\rangle = {} & 2\langle 1s|O^1|1s\rangle \\
+ {} & \langle 2s|O^1|2s\rangle \\
+ {} & \langle 3d^{+2}|O^1|3d^{+2}\rangle \\
+ {} & \langle 1s1s|O^2|1s1s\rangle \\
+ {} & 2\langle 1s2s|O^2|1s2s\rangle \\
- {} & \langle 1s2s|O^2|1s2s\rangle \\
+ {} & 2\langle 1s3d^{+2}|O^2|1s3d^{+2}\rangle \\
- {} & \langle 1s3d^{+2}|O^2|3d^{+2}1s\rangle \\
+ {} & \langle 2s3d^{+2}|O^2|2s3d^{+2}\rangle \\
+ {} & \langle 1s1s2s|O^3|1s1s2s\rangle \\
- {} & \langle 1s1s2s|O^3|2s1s1s\rangle \\
+ {} & \langle 1s1s3d^{+2}|O^3|1s1s3d^{+2}\rangle \\
- {} & \langle 1s1s3d^{+2}|O^3|1s3d^{+2}1s\rangle \\
+ {} & 2\langle 1s2s3d^{+2}|O^3|1s2s3d^{+2}\rangle
\end{aligned}$$

$$- \quad \langle 1s2s3d^{+2}|O^3|2s1s3d^{+2}\rangle$$

$$- \quad \langle 1s2s3d^{+2}|O^3|3d^{+2}2s1s\rangle$$

$$+ \quad 2\langle 1s2s3d^{+2}|O^3|1s3d^{+2}2s\rangle$$

$$- \quad 2\langle 1s2s3d^{+2}|O^3|3d^{+2}1s2s\rangle$$

$$+ \quad \langle 1s1s2s3d^{+2}|O^4|1s1s2s3d^{+2}\rangle$$

$$- \quad \langle 1s1s2s3d^{+2}|O^4|1s3d^{+2}2s1s\rangle$$

$$- \quad \langle 1s1s2s3d^{+2}|O^4|2s1s1s3d^{+2}\rangle$$

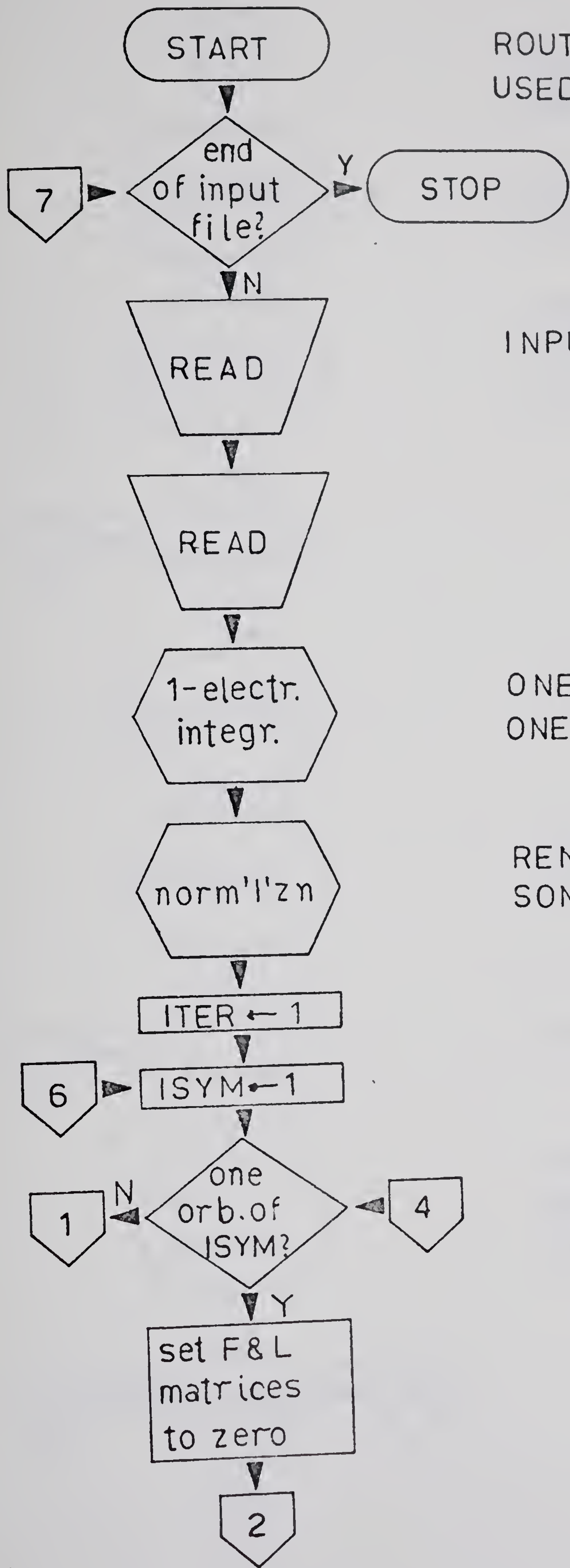
$$+ \quad 2\langle 1s1s2s3d^{+2}|O^4|2s3d^{+2}1s1s\rangle$$

$$+ \quad \langle 1s1s2s3d^{+2}|O^4|1s1s3d^{+2}2s\rangle$$

$$- \quad 2\langle 1s1s2s3d^{+2}|O^4|1s3d^{+2}1s2s\rangle$$

THE PROGRAM.

It seems to be extraordinarliy difficult to describe intelligently a computer program of some complexity by words. We will instead present the logical flow of the program in form of a flow diagram. This diagram with its annotations and with the listing of the program at the end of this appendix should facilitate the understanding and use of the program. After the flow diagram a section is concerned with the listing of all routines that have been used, except the routines which are part of publicly available libraries. This listing carries short annotations as to where and how the routines are employed and it is arranged in nearly the same order as the listing of the programs. At the end programs which have been used in preparing this thesis have been listed.



ROUTINES USED — REMARKS

SCF-PROGRAM

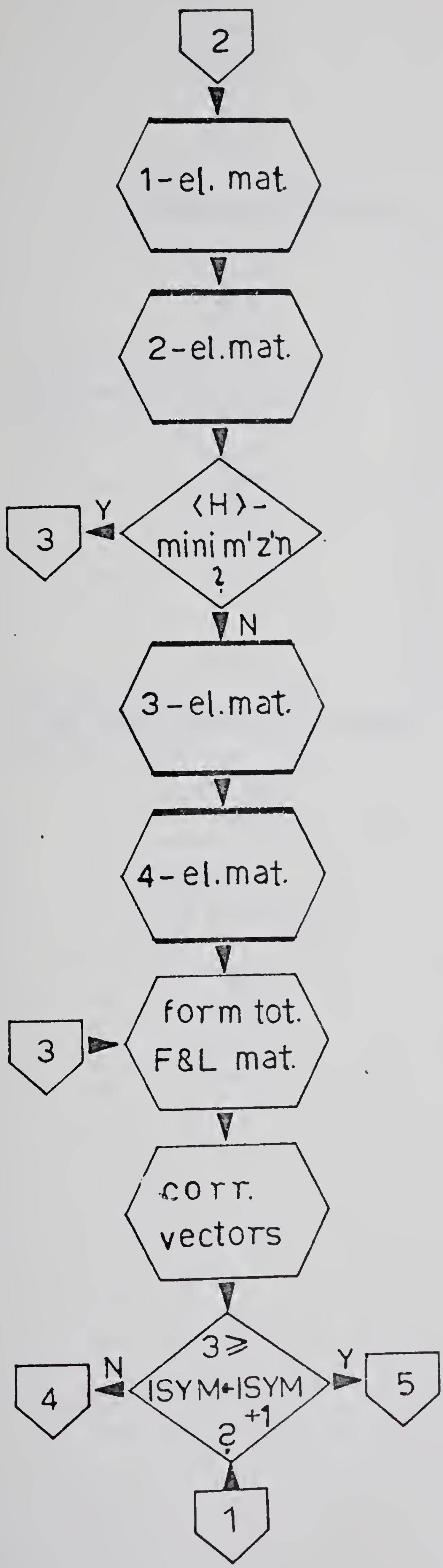
INPUT

no. of orbitals
no. of basis fns.
method, charge
energy, orbital exp.
starting vectors
INTEGRALS

ONEINT
ONEI

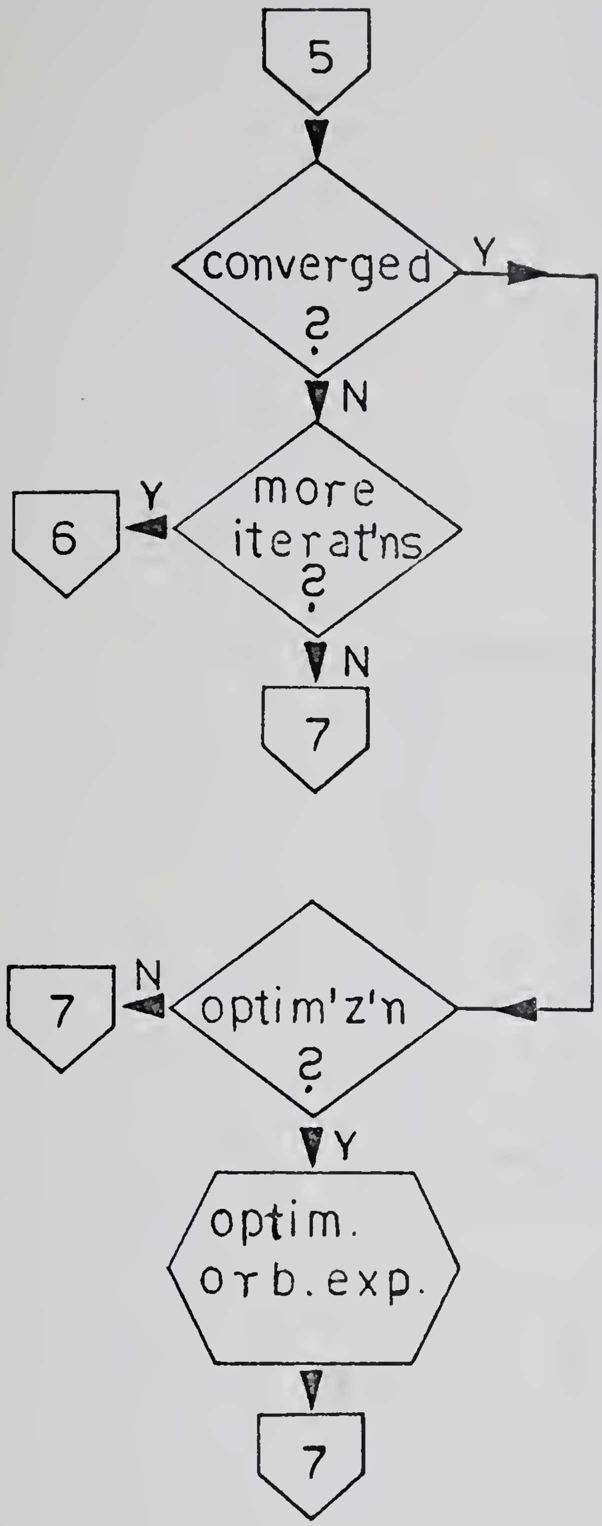
RENORM
SOMS

ISYM:
1 s-orbital
2 p- "
3 d- "

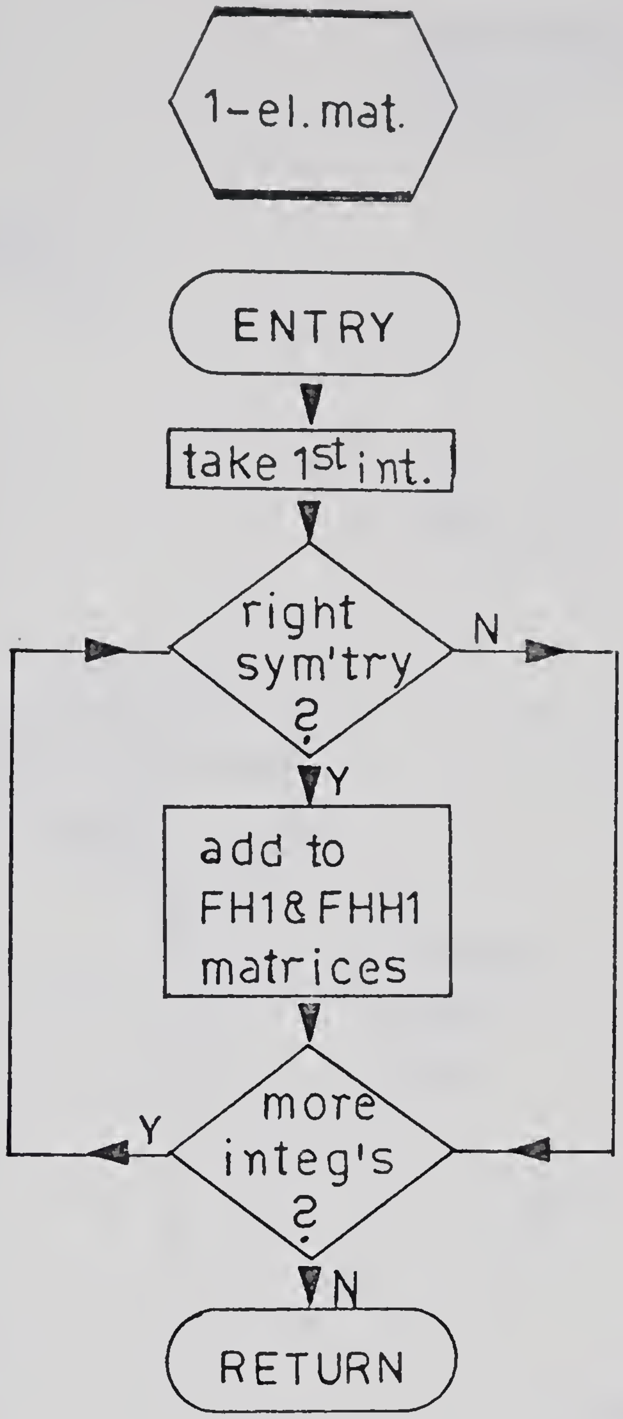


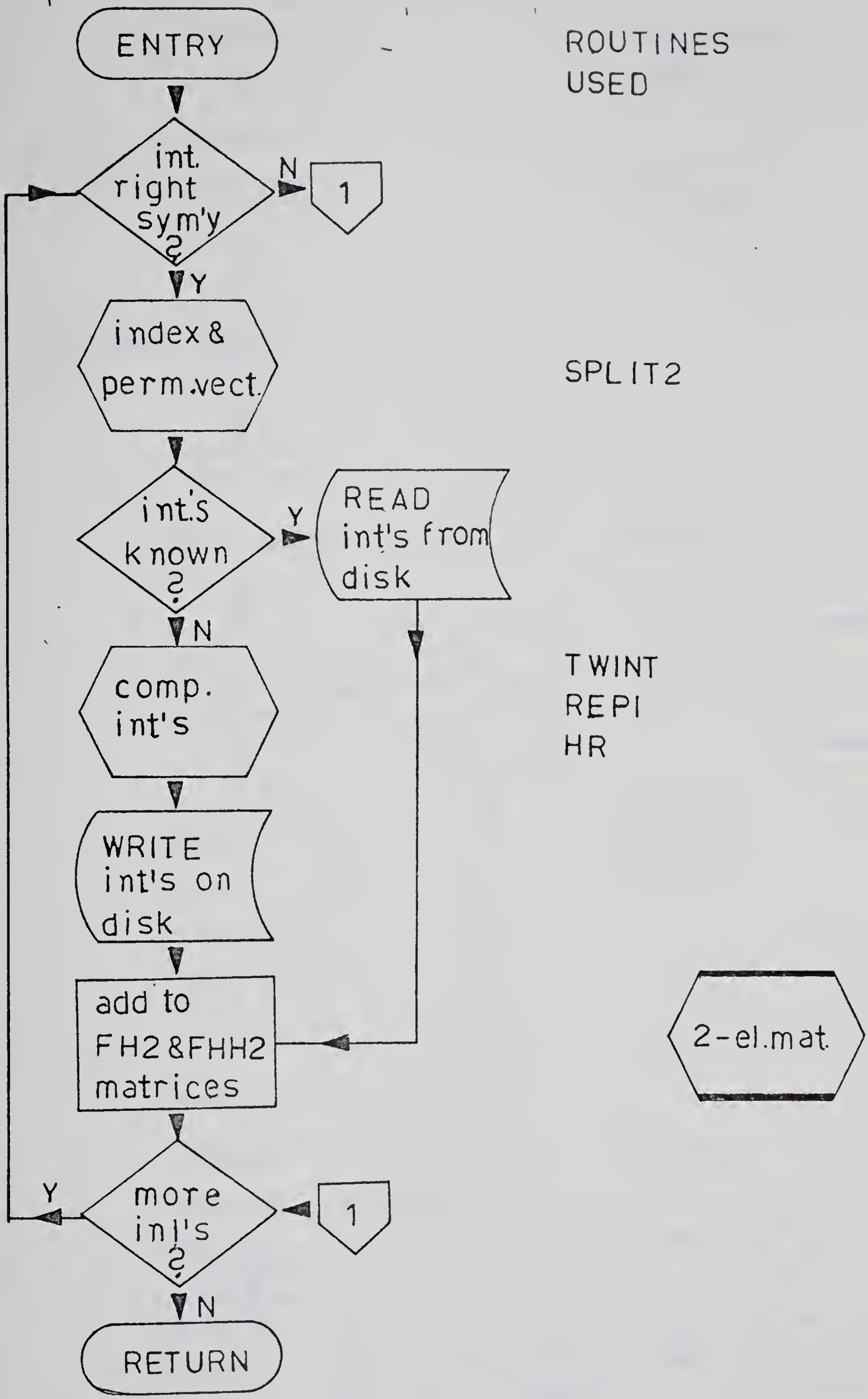
ROUTINES USED	REMARKS
ONEEL SYMCHE	thickly outlined routines are diagrammed separately
TWOEL TWINT SPLIT2	
THREEL	
FOUREL	
COMBIN	
HINZE SOLVER GAUSS	

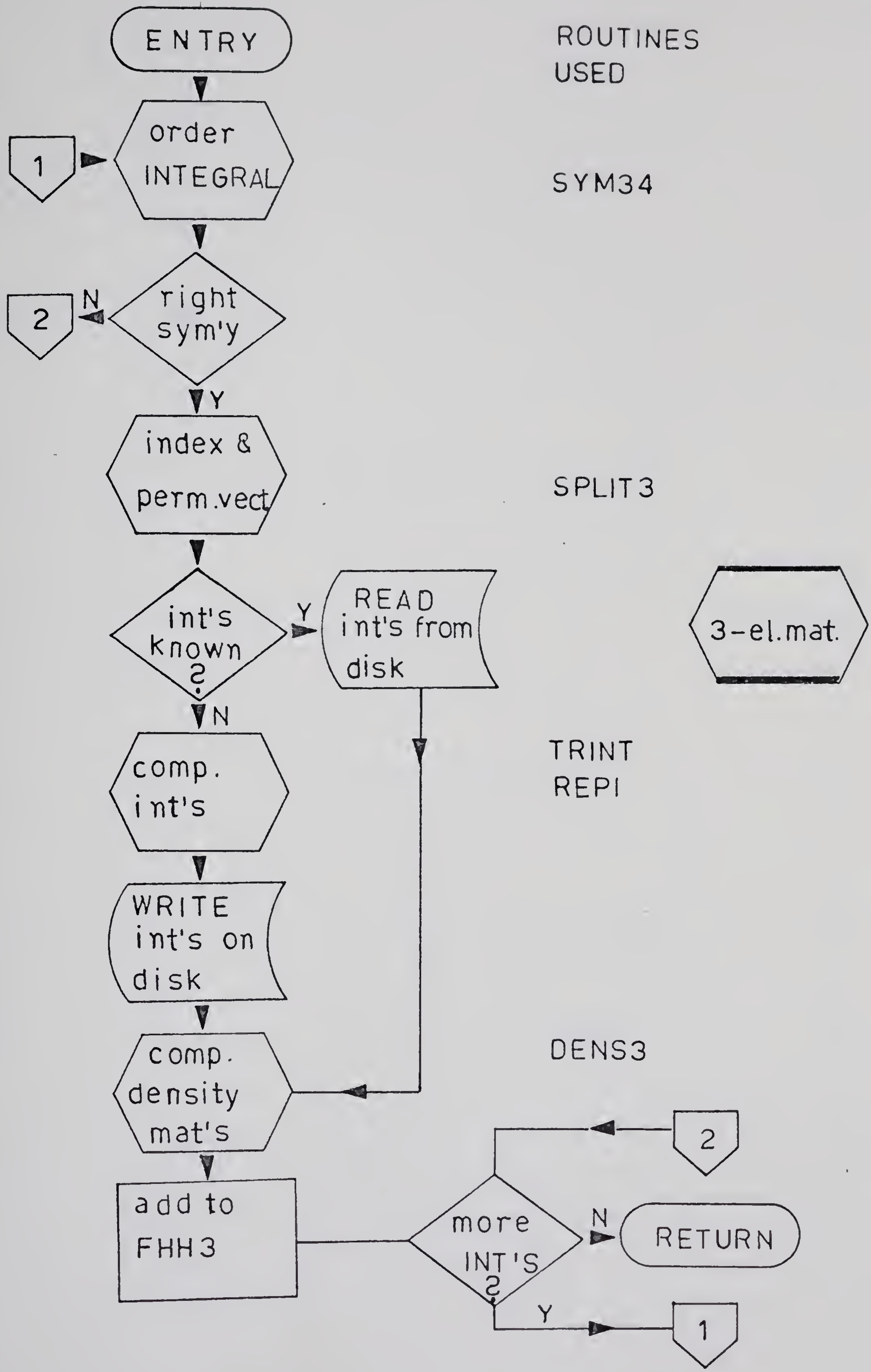
ROUTINES USED REMARKS



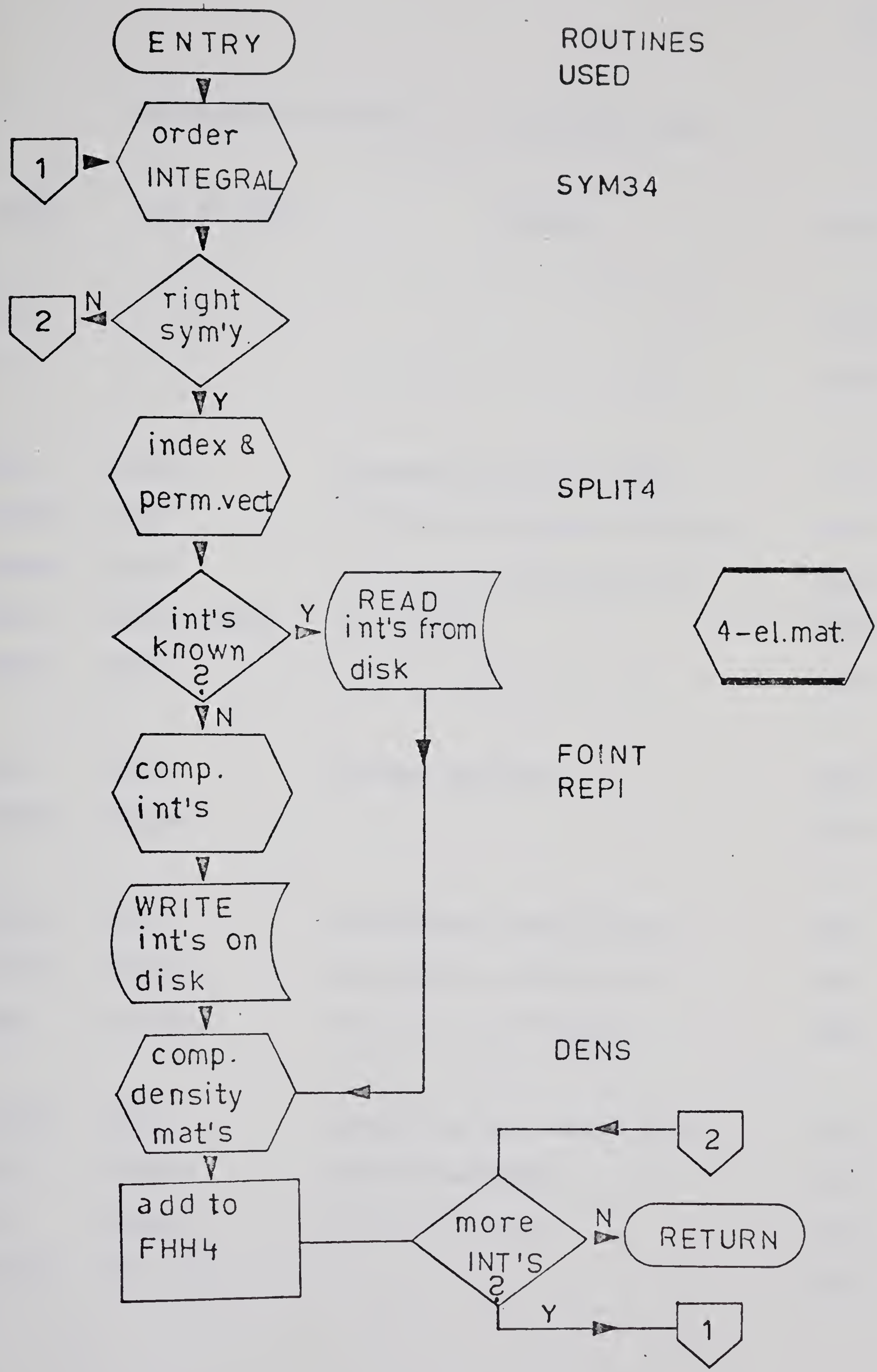
OPTIM
CHANGE
POLYNO
SCFCYC


1-el. mat.
ENTRY
take 1st int.
right sym'try ?
N
Y
add to FH1 & FHH1 matrices
more integ's ?
Y
N
RETURN


ENTRY
ROUTINES USED
int. right sym'y ?
N
1
Y
index & perm.vect.
SPLIT2
int's known ?
Y
READ int's from disk
N
comp. int's
TWINT
REPI
HR
WRITE int's on disk
add to FH2 & FHH2 matrices
2-el.mat.
Y
more int's ?
1
N
RETURN



ENTRY
order INTEGRAL
1
right sym'y
N
2
Y
index & perm.vect
int's known ?
Y
READ int's from disk
N
comp. int's
WRITE int's on disk
comp. density mat's
add to FHH3
2
more INT'S ?
N
RETURN
Y
1
ROUTINES USED
SYM34
SPLIT3
3-el.mat.
TRINT
REPI
DENS3


ENTRY
ROUTINES USED
order INTEGRAL
1
SYM34
right sym'y
2
N
Y
index & perm.vect
SPLIT4
int's known ?
Y
READ int's from disk
4-el.mat.
N
comp. int's
FOINT
REPI
WRITE int's on disk
comp. density mat's
DENS
2
add to FHH4
more INT'S ?
N
RETURN
Y
1

ROUTINES TO COMPUTE L-S EIGENFUNCTIONS

NAME	CALLED FROM	PURPOSE	CODE
MAIN			LSQH
			LSQT
VECT	MAIN	Routines to set up the	VECT
EXPAND	MAIN	different Slators belonging	EXPD
DETVAR	MAIN	to a given configuration	DETV
RESET	MAIN, CHECK	and state.	RSET
CHECK	MAIN		CHCK
OUT	LOP	Output routines	OUTP
SHREIB	OPERAT		SHRB
OPERAT	MAIN	Subroutines determining	OPE
LSSQUA	OPERAT	the matrix elements of	LSS
COMP	LSSQUA	$\langle \phi \mid L^2 + k*S^2 \mid \phi \rangle$	COMP
LMINUS	MAIN	Computing the states MS-1,	LMIN
SOP	LMINUS	ML-1 from MS,ML.	S-OP
LOP	LMINUS		L-OP
SEARCH	SOP, LOP		SRCH

NAME	CALLED FROM	PURPOSE	CODE
INTCOE	MAIN	The Slators are compared	INTH
			INTT
FILL	MAIN, LOP	for each value of ML and	FILH
	LMINUS		FILT
SORT	INTCOE	MS and the 1,2,3, and 4	SRTH
			SRTT
COMP1	INTCOE	electron integrals to be	COMH
			COMT
ONE	SORT	used as input for the	ONEH
			ONET
TWO	SORT	wave-function routine	TWOH
			TWOT
THREE	SORT	are computed.	THRH
			THRT
FOUR	SORT		FORH
			FORT
SIG	FUNCTION	Determines if a permutation is odd or even.	FSIG
DEIGE	OPERAT	Jacobi diagonalization IBM-SSP-Routine.	

ROUTINES TO COMPUTE SCF WAVEFUNCTIONS

NAME	CALLED FROM	PURPOSE	CODE
MAIN			MAIN
INPUT	MAIN	Reads in the starting vectors, integrals, and indicates the minimization to be done.	INPT
ONEINT	MAIN	Sets up one electron integrals	ONEI
RENORM	MAIN	Normalizes the vectors.	NORM
ONEEL	MAIN	Sets up one electron matrices.	ONEE
TWOEL	MAIN	2-electron-matrices.	TWOE
TWINT	TWOELE	2-electron-integrals.	TWIN
THREEL	MAIN	3-electron-matrices.	THRE
TINT3	THREEL	3-electron-intrgrals.	TINT
DENS3	THREEL	3-electron-density matrices.	DNS3

[illegible]

NAME	CALLED FROM	PURPOSE	CODE
FOUREL	MAIN	4-electron-matrices.	FOUR
FOINT	FOUREL	4-electron-integrals.	FOIN
DENS	FOUREL	4-electron-density matrices.	DNS4
LIES	DENS	Reads in the 2-electron-integrals required in FOUREL	LIES
DIAGO	MAIN	Diagonalization of 4-electron-matrices.	DIAG
COMBIN	MAIN	Combines the F-matrices according to which minimization is desired.	COMB
CNVRGC	MAIN	Checks if vectors converge.	CONV
AITKEN	MAIN	Aitken-Delta-Acceleration.	AITK
uts|PRPRTS	MAIN	Computs value of $\langle 1/r \rangle$ etc.	PROP
HINZE	MAIN	Combines the L and F matrices so that G-supermatrix and G supervector for computing c are obtained.	HINZ
SOLVER	HINZE	Gaussian elimination with	SOLV
GAUS	SOLVER	pivoting of row and columns.	GAUS

NAME	CALLED FROM	PURPOSE	CODE
ENER	MAIN, COMBIN	Computes <H> and <H >	ENER
EXHH	MAIN, COMBIN	respectively.	EXHH
OPTIM	MAIN	Optimization routine.	OPTI
CHANGE	OPTIM	Aiding optimization.	CHNG
POLYNO	OPTIM	Aiding optimization.	POLY
SCFCYC	OPTIM	Aiding optimization.	SCFC
OUT0	MAIN	Various output routines.	OUT1
OUT01	MAIN		OUT1
OUT1	MAIN		OUT1
OUT2	MAIN		OUT1
OUT3	MAIN		OUT1
OUT4	MAIN		OUT1
OUT5	MAIN		OUT1
OUTPUT	MAIN		OUT2

NAME	CALLED FROM	PURPOSE	CODE
SPLIT2	TWOEL	All these routines reorder	SPLI
SPLIT3	SYM34	the input-integrals so that	SPLI
SPLIT4	SYM34	the indices of the expansion	SPLI
SYMAS1	SPLIT2, SPLIT3 SPLIT4	vectors and the integrals over the Slater functions	SYA1
SYMCHE	TWOEL, SYM34	coincide.	SYCH
SYMAS2	SPLIT2,SPLIT3, SPLIT4		SYA2
SYMAS3	SPLIT3, SPLIT4, TWINT		SYA3
IDNOM	SPLIT2, SPLIT3, SPLIT4		IDNO
SYM34	THREEL, FOUREL		SYA3

Routines Not Programmed by the Author.

NAME	PURPOSE	CODE
	(See also the table with SYSTEM-SUBROUTINES.)	
LOGIOU	Direct access routine Author Larry Thiel, Computing Centre University of Alberta.	LIOU
DEIGE	Jacobi diagonalization Author IBM/SSP	
SOMS	Schmidt-orthogonalization	SOMS
MULTS	Schmidt matrix multiplication	MULT
VMULT	subroutines.	VMUL
ONEI	Slater function integral routines	ONIN
HR		HRIN
REPI		REPI
ANGLI		ANGI
UF		ANGI
VF		ANGI
FIDA		ANGI
FIDB		ANGI
ENMI	All programmes in this section by F. W. Birss.	ENMI

SYSTEM (MTS) ROUTINES

NAME	PURPOSE
READ	e.g. CALL READ(INTEG,LEN,0,LNR,2, 100)
WRITE	Used to read and write integrals and density matrices from or to disk.
LOGIOU POINT NOTE	Used to determine the parameters that allow access to sepuential files stored on disk.
REWIND	Used to reset the sequential file used for storing the Density matrices in each iteration.
TIME	To time the execution of the program

Routines READ and WRITE are described in FORTRAN G and H MANUAL, May 1970, University of Alberta, Computing Centre.

Routines REWIND and TIME are described in SUBROUTINE LIBRARIES MANUAL, October 1970, University of Alberta, Computing Centre.

Routines NOTE and POINT are described in SYSTEM SUBROUTINE MANUAL, June 1970, University of Alberta, Computing Centre.

```
      REAL*8 LSSQMA(52,52)/2704*0.D0/,EIGVAL(52,52),EIGVEC(52,52),
     1B(52),F1,F2,SPIN(22)
      EQUIVALENCE(LSSQMA(1),EIGVAL(1))
C
C  THIS ROUTINE IS SET UP TO CALCULATE LS-EIGENFUNCTIONS OF UP
C  TO TEN ELECTRONS. TO INCLUDE A LARGER NUMBER OF ELECTRONS
C  SEEMS SENSELESS, SINCE RUSSELL-SAUNDERS COUPLING BREAKS DOWN
C
C  THE EIGENVALUES ARE ON THE AVERAGE ACCURATE TO 11 SIGNIFICANT
C  FIGURES. IF HIGHER ACCURACY IS DESIRED, CHANGE STATEMENT 5 IN
C  SUBROUTINE 'DEIGE'.
C
C
C  THE ROUTINE CAN HANDLE STATES WHICH ARE REPRESENTED BY UP TO
C  52 SLATERDETERMINANTS. IF A LARGER NUMBER OF SLATORS ARISE, THE
C  FOLLOWING CHANGES HAVE TO BE MADE
C     CHANGE THE DIMENSIONS OF LSSSQMA,EIGVAL,EIGVEC,B,SLDV,NUMDET
C     IN THE MAIN PROGRAM.
C     CHANGE THE FORMAT STATEMENTS IN THE SUBROUTINE SHREIB
C     CHANGE DIMENSION OF SLDV,NUMDET IN SUBROUTINES
C       DETVAR
C       OPERAT
C       LSSQUA
C       COMP
C       OUTPU
C
       INTEGER*2  DMAT(4,100), CONFIG ( 33), STATE (2), IVEC (20),
     1ICOMV (20), ISTA (20), SLDV ( 52,4,20),NUMDET(52,20),CMAT(4,20)
     3,LINE(22),STTE(2)
CINPUT IS AS FOLLOWS:
C  M   THE NUMBER OF UNEQUIVALENT STATES TIMES 3
C  N   THE NUMBER OF ELECRONS
CCONFIGURATION: 1S1 2P2 3D1    =  01 00 01 02 01 02 03 02 01
CSTATE           3P            =  03 01
C  PUT  M,N   CONF,STATE,   AS CONTINUOUS I4 INPUT
CIVEC CONTAINS THE POSITION FOR SLD IN DMAT
CICOMPV CONTAINS THE MAXIMUM IVEC CAN REACH
C
C   AFTER THE STATE WITH ML=L AND MS=S HAS BEEN COMPUTED, L- & S- ARE
C   APPLIED(IN SUBROUTINE LMINUS) TO OBTAIN ALL POSSIBLE E'FNS FOR ALL
C   VALUES OF ML AND MS.
C
C   THE OUTPUT IS WRITTEN ON UNIT(6)
C
C   INTCOE COMPUTES THE INTEGRALS OBTAINED BY OPERATING WITH ONE-,
C   TWO-,THREE-,AND FOUR-ELECTRON-OPERATORS
C   IT WRITES THE RESULTS ON UNIT(8)
C
    1 READ(5,901,END=23)M,N,(CONFIG(J1),J1=1,M),(STATE(J1),J1=1,2)
      DO 2 J1 = 1,52
      DO 2 J2 = 1,52
    2 LSSQMA(J1,J2) = 0.D0
      CALL VECT(IVEC,ISTA,ICOMV,M,CONFIG,N)
      CALL EXPAND (STATE,CONFIG,DMAT,N,M)
C AFTER DMAT IS COMPUTED IT'S COLUMNS ARE USED TO SET UP ALL POSSIBLE
C SLATERDETERMINANTS, WHICH ARE CHECKED IF THEY FULFILL THE STATE COND.
      K=0
      I1=N
    9 CALL DETVAR (DMAT,IVEC,SLDV,N,STATE,K,NUMDET,&45)
      IVEC(I1) = IVEC (I1) + 1
```

```
      IF (IVEC (I1) .LT. ICONV(I1)) GO TO 9                              LSQH
      CALL RESET (IVEC,I1,&19,&29,N,ISTA,&19)                            LSQH
   19 CALL CHECK (IVEC,ISTA,&9 ,N ,&29,ICONV)                            LSQH
   29 STTE(1) = STATE(1)                                                 LSQH
      STTE(2) = STATE(2)                                                 LSQH
      CALL OUTPU (SLDV,STATE,CONFIG,K,N,&45,0,STTE)                      LSQH
      CALL OPERAT (SLDV,LSSOMA,K,STATE,CMAT,N,EIGVEC,P,&45,I1)           LSQH
      CALL FILL(SLDV,EIGVEC,K,N,I1)                                      LSQH
      CALL INTCOE(N,K,I1)                                                LSQH
   45 GO TO 1                                                            LSQH
  901 FORMAT (20 I 4)                                                    LSQH
   23 STOP                                                               LSQH
      END                                                                LSQH
      SUBROUTINE FILL (SL,EIGVEC,K,NOE,I1)                               FILH
C                                                                        FILH
C  THIS ROUTINE COMPRESSES THE SLATORS FROM 4-ON TO 1-ON                 FILH
C                                                                        FILH
C  SLATOR(*,*,*) CONTAINS THE COMPRESED INDEX CALCULATED                 FILH
C  FROM SL(*,*,*)                                                        FILH
C  N                                                                     FILH
C  L                                                                     FILH
C  ML  ARE SELFEXPLANATORY                                               FILH
C  MS                                                                    FILH
C                                                                        FILH
      COMMON/FINT/LVEC(3,52),SLATOR(52,10)                               FILH
      REAL*8 EIGVEC(K,K),LVEC                                            FILH
      INTEGER*2 L(52,4,10)                                              FILH
      INTEGER SLATOR                                                     FILH
      INDG(N) = (N-2)*9-3                                                FILH
      INDF(L,M) = L+L*L+M                                                FILH
      DO 10 J10 = 1,K                                                    FILH
      DO 10 J11 = 1,NOE                                                  FILH
      N = SL(J10,1,J11)                                                  FILH
      L = SL(J10,2,J11)                                                  FILH
      ML= SL(J10,3,J11)                                                  FILH
      MS= SL(J10,4,J11)                                                  FILH
      LML = INDF (L,ML)                                                  FILH
      IF (N .LE. 2) GO TO 1                                              FILH
      INCOMP = (LML+INDG(N))*MS                                          FILH
      GO TO 10                                                           FILH
    1 INCOMP = (LML+N)*MS                                                FILH
   10 SLATOR (J10,J11) = INCOMP                                          FILH
      J22=K-I1                                                           FILH
      DO 20 J20=1,I1                                                     FILH
      DO 20 J21=1,K                                                      FILH
    0 LVEC(J20,J21)=EIGVEC(J21,J22+J20)                                  FILH
      RETURN                                                             FILH
      END                                                                FILH
      SUBROUTINE INTCOE(NOE,K,I1)                                        INTH
C PURPOSE:                                                               INTH
C     TO COMPUTE SYMBOLICALLY THE INTEGRALS WHICH ARE OBTAINED WHEN      INTH
C     L-S-EIGENSTATES                                                    INTH
C     VARIABLES:                                                         INTH
C     TERM: THE TERMSYMBOL, EQUIVALENT TO STATE IN 'LSQ'                 INTH
C     INT*: ARRAYS IN WHICH THE SYMBOLIC FORM OF THE INTEGRALS IS STOREDINTH
C     FAC*: ARRAYS IN WHICH THE COMPUTE COEFFICIENTS ARE STORED          INTH
      IMPLICIT REAL*8 (A-H,O-Z)                                          INTH
      COMMON/FINT/LVEC(3,52),SLATOR(52,10)                               INTH
      COMMON/SUBINT/FAC(3),FAC1(3,50),FAC2(3,100),FAC3(3,200),FAC4(3,300INTH
     .),INT1(50),INT2(4,100),INT3(6,200),INT4(8,300),SLASHO(2,10),DIFORBINTH
```

```
   .(4)                                                                    INTH
    INTEGER SLATOR,SLASHO,DIFORB                                           INTH
    REAL*8 LVEC                                                            INTH
    INT1(1)=0                                                              INTH
    INT2(1,1)=0                                                            INTH
    INT3(1,1)=0                                                            INTH
    INT4(1,1)=0                                                            INTH
    LIM1=0                                                                 INTH
    LIM2=0                                                                 INTH
    LIM3=0                                                                 INTH
    LIM4=0                                                                 INTH
    WRITE(8,910)                                                           INTH
)   FORMAT('1')                                                            INTH
    WRITE(8,908) ((SLATOR(IA1,IA2),IA2=1,NOE),IA1=1,K)                     INTH
    WRITE(8,909) ((LVEC(IA1,IA2),IA2=1,K),IA1=1,I1)                        INTH
;   FORMAT(' ',20I4)                                                       INTH
)   FORMAT(' ',10D12.4)                                                    INTH
    IF(I1.GT.3) GO TO 11                                                   INTH
    DO 1 JB=1,K                                                            INTH
    DO 1 JC=JB,K                                                           INTH
    CALL COMP1(JB,JC,I1,ICODE,&1,NOE)                                      INTH
    CALL SORT(ICODE,LIM1,LIM2,LIM3,LIM4,NOE,I1                            INTH
  1 CONTINUE                                                               INTH
    WRITE(8,900)LIM1,LIM2,LIM3,LIM4                                        INTH
    DO 7 JA=1,LIM1                                                         INTH
  7 WRITE(8,903) INT1(JA),(FAC1(IA,JA),IA=1,I1)                            INTH
    DO 8 JA=1,LIM2                                                         INTH
  8 WRITE(8,904)(INT2(IA,JA),IA=1,4,(FAC2(IB,JA),IB=1,I1)                 INTH
    IF(NOE.LT.3)RETURN                                                     INTH
    DO 9 JA = 1,LIM3                                                       INTH
  9 WRITE(8,905) (INT3(IA,JA),IA=1,6),(FAC3(IA,JA),IA=1,I1)                INTH
    IF(NOE.LT.4)RETURN                                                     INTH
    DO 10 JA=1,LIM4                                                        INTH
 10 WRITE(8,906) (INT4(IA,JA),IA=1,8),(FAC4(IA,JA),IA=1,I1)                INTH
    RETURN                                                                 INTH
 11 WRITE(8,907)                                                           INTH
    STOP                                                                   INTH
900 FORMAT(20I4)                                                           INTH
901 FORMAT(3D26.18)                                                        INTH
902 FORMAT(2I4)                                                            INTH
903 FORMAT(33X,I3,3D25.15)                                                 INTH
904 FORMAT(18X,2(3X,2I3),3D25.15)                                          INTH
905 FORMAT(9X,3(3X,2I3),3D25.15)                                           INTH
906 FORMAT(4(3X,2I3),3D25.15)                                              INTH
907 FORMAT('0',131('*')/40X,'MORE THAN THREE LINEARLY INDEPENT EIGENFUINTH
   1NCTIONS'/131('*'))                                                     INTH
    END                                                                    INTH
    SUBROUTINE COMP1(I,J,I1,ICODE,*,NOE)                                   COMH
    IMPLICIT REAL*8 (A-H,O-Z)                                              COMH
    COMMON/FINT/LVEC(3,52),SLATOR(52,10)                                   COMH
    COMMON/SUBINT/FAC(3),FAC1(3,50),FAC2(3,100),FAC3(3,200),FAC4(3,300COMH
   .),INT1(50),INT2(4,100),INT3(6,200),INT4(8,300),SLASHO(2,10),DIFORBCOMH
   .(4)                                                                    COMH
    INTEGER SLATOR,SLASHO,DIFORB                                           COMH
    REAL*8 LVEC                                                            COMH
    ISUM=0                                                                 COMH
    ICODE=0                                                                COMH
    FACT=1.D0                                                              COMH
 11 DO 1 JA=1,NOE                                                          COMH
    SLASHO(1,JA)=SLATOR(I,JA)                                              COMH
```

```
1 SLASHO(2,JA)=SLATOR(J,JA)                                          COMH
  IF (J.EQ.I) GO TO 3                                               COMH
  DO 5 JA=1,NOE                                                     COMH
  DO 6 JB=1,NOE                                                     COMH
  IF(SLASHO(1,JA).EQ.SLASHO(2,JB)) GO TO 2                          COMH
6 CONTINUE                                                          COMH
  ICODE=ICODE+1                                                     COMH
  DIFORB(ICODE)=JA                                                  COMH
  IF (ICODE.GT.4) RETURN1                                           COMH
  GO TO 5                                                           COMH
2 IF(JA.EQ.JB) GO TO 5                                              COMH
  ISUM = ISUM+1                                                     COMH
  IEX=SLASHO(2,JA)                                                  COMH
  SLASHO(2,JA) = SLASHO(2,JB)                                       COMH
  SLASHO(2,JB)=IEX                                                  COMH
5 CONTINUE                                                          COMH
  FACT=2.D0                                                         COMH
3 DO 7 JA=1,I1                                                      COMH
7 FAC(JA)=LVEC(JA,I)*LVEC(JA,J)*DFLOAT((-1)**ISUM)*FACT             COMH
  IF(ICODE.EQ.0) ICODE=1                                            COMH
  RETURN                                                            COMH
  END                                                               COMH
  SUBROUTINE SORT(ICODE,LIM1,LIM2,LIM3,LIM4,NOE,I1)                 SRTH
  IMPLICIT REAL*8 (A-H,O-Z)                                         SRTH
  COMMON/FINT/LVEC(3,52),SLATOR(52,10)                              SRTH
  COMMON/SUBINT/FAC(3),FAC1(3,50),FAC2(3,100),FAC3(3,200),FAC4(3,300SRTH
 .),INT1(50),INT2(4,100,INT3(6,200),INT4(8,300),SLASHO(2,10),DIFORBSRTH
 .(4)                                                               SRTH
  INTEGER SLATOR,SLASHO,DIFORB                                      SRTH
  REAL*8 LVEC                                                       SRTH
  GO TO (1,2,3,4),ICODE                                             SRTH
1 DO 7 JA=1,NOE                                                     SRTH
  CALL ONE (JA,LIM1,I1)                                             SRTH
  JBL=JA+1                                                          SRTH
  IF (JBL.GT.NOE)GOTO 7                                             SRTH
  DO 6 JB=JBL,NOE                                                   SRTH
  CALL TWO(JA,JB,LIM2,I1)                                           SRTH
  IF(NOE.LT.3) GO TO 6                                              SRTH
  JCL=JB+1                                                          SRTH
  IF(JCL.GT.NOE)GOTO 6                                              SRTH
  DO 5 JC=JCL,NOE                                                   SRTH
  CALL THREE(JA,JB,JC,LIM3,I1)                                      SRTH
  IF (NOE.LT.4) GO TO 5                                             SRTH
  JDL=JC+1                                                          SRTH
  IF(JDL.GT.NOE)GOTO 5                                              SRTH
  DO 8 JD=JDL,NOE                                                   SRTH
  CALL FOUR(JA,JB,JC,JD,LIM4,I1)                                    SRTH
8 CONTINUE                                                          SRTH
5 CONTINUE                                                          SRTH
6 CONTINUE                                                          SRTH
7 CONTINUE                                                          SRTH
  RETURN                                                            SRTH
2 JA=DIFORB(1)                                                      SRTH
  JB=DIFORB(2)                                                      SRTH
  CALL TWO(JA,JB,LIM2,I1)                                           SRTH
  IF(NOE.LT.3) RETURN                                               SRTH
  DO 9 JC=1,NOE                                                     SRTH
  IF((JC.EQ.JA).OR.(JC.Q.JB)) GO TO 9                              SRTH
  CALL THREE(JA,JB,JC,LIM3,I1)                                      SRTH
  IF(NOE.LT.4) GO TO 9                                              SRTH
```

```
      JDL=JC+1                                                          SRTH
      IF(JDL.GT.NOE)GO TO 9                                             SRTH
      DO 10 JD=JDL,NOE                                                  SRTH
      IF((JD.EQ.JA).OR.(JD.EQ.JB).OR.(JD.EQ.JC)) GO TO 10               SRTH
      CALL FOUR(JA,JB,JC,JD,LIM4,I1)                                    SRTH
   10 CONTINUE                                                          SRTH
    9 CONTINUE                                                          SRTH
      RETURN                                                            SRTH
    3 JA=DIFORB(1)                                                      SRTH
      JB=DIFORB(2)                                                      SRTH
      JC=DIFORB(3)                                                      SRTH
      CALL THREE(JA,JB,JC,LIM3,I1)                                      SRTH
      IF(NOE.LT.4) RETURN                                               SRTH
      DO 11 JD=1,NOE                                                    SRTH
      IF((JD.EQ.JA).OR.(JD.EQ.JB).OR.(JD.EQ.JC))GOTO 11                 SRTH
      CALL FOUR(JA,JB,JC,JD,LIM4,I1)                                    SRTH
   11 CONTINUE                                                          SRTH
      RETURN                                                            SRTH
      IDF1=DIFORB(1)                                                    SRTH
      IDF2=DIFORB(2)                                                    SRTH
      IDF3=DIFORB(3)                                                    SRTH
      IDF4=DIFORB(4)                                                    SRTH
      CALL FOUR(IDF1,IDF2,IDF3,IDF4,LIM4,I1)                            SRTH
      RETURN                                                            SRTH
      END                                                               SRTH
      SUBROUTINE ONE(I,LIM1,I1)                                         ONEH
      IMPLICIT REAL*8 (A-H,O-Z)                                         ONEH
      COMMON/FINT/LVEC(3,52),SLATOR(52,10)                              ONEH
      COMMON/SUBINT/FAC(3),FAC1(3,50),FAC2(3,100),FAC3(3,200),FAC4(3,300ONEH
     .),INT1(50),INT2(4,100),INT3(6,200),INT4(8,300),SLASHO(2,10),DIFORBONEH
     .(4)                                                               ONEH
      INTEGER SLATOR,SLASHO,DIFORB                                      ONEH
      REAL*8 LVEC                                                       ONEH
      KB=IABS(SLASHO(1,I))                                              ONEH
      DO 1 JA=1,LIM1                                                    ONEH
      LI=JA                                                             ONEH
      IF (INT1(JA).EQ.KB) GO TO 2                                       ONEH
    1 CONTINUE                                                          ONEH
      LIM1=LIM1+1                                                       ONEH
      IF (50.LT.LIM1) GO TO 3                                           ONEH
      INT1(LIM1)=KB                                                     ONEH
      DO 4 JA=1,I1                                                      ONEH
    4 FAC1(JA,LIM1)=FAC(JA)                                             ONEH
      RETURN                                                            ONEH
    2 DO 5 JA=1,I1                                                      ONEH
    5 FAC1(JA,LI)=FAC1(JA,LI)+FAC(JA)                                   ONEH
      RETURN                                                            ONEH
    3 WRITE(8,900)                                                      ONEH
      STOP                                                              ONEH
  900 FORMAT('0',131('*')/'MORE THAN 50 ONE-ELE INTEGRALS'/131('*'))    ONEH
      END                                                               ONEH
      SUBROUTINE TWO(II,JJ,LIM2,I1)                                     TWOH
      IMPLICIT REAL*8 (A-H,O-Z)                                         TWOH
      COMMON/FINT/LVEC(3,52),SLATOR(52,10)                              TWOH
      COMMON/SUBINT/FAC(3),FAC1(3,50),FAC2(3,100),FAC3(3,200),FAC4(3,300TWOH
     .),INT1(50),INT2(4,100),INT3(6,200),INT4(8,300),SLASHO(2,10),DIFORBTWOH
     .(4)                                                               TWOH
      INTEGER SLATOR,SLASHO,DIFORB,IV(2)                                TWOH
      REAL*8 LVEC                                                       TWOH
      INTEGER IX(4,4)/1,2,3,4,3,4,1,2,2,1,4,3,4,3,2,1/                  TWOH
```

```
      LOGICAL SPIN                                                      TWOH
      SPIN(I,J,K,L)=(0.GT.ISIGN(1,I)*ISIGN(1,J)).OR.(0.GT.ISIGN(1,K)*   TWOH
     1ISIGN(1,L))                                                       TWOH
      I=II                                                              TWOH
      J=JJ                                                              TWOH
      I1B=IABS(SLASHO(1,I))                                             TWOH
      I2B=IABS(SLASHO(1,J))                                             TWOH
      SIGN=1.D0                                                         TWOH
      DO 1 JA=1,2                                                       TWOH
      DO 2 JB=1,2                                                       TWOH
      IF(JB.EQ.JA) GO TO 2                                              TWOH
      IF(JB.EQ.1) SIGN=-SIGN                                            TWOH
      IV(JA)=I                                                          TWOH
      IV(JB)=J                                                          TWOH
      IF(SPIN(SLASHO(1,I),SLASHO(2,IV(1)),SLASHO(1,J),SLASHO(2,IV(2)))  TWOH
     1)GO TO 2                                                          TWOH
      I1K=IABS(SLASHO(2,IV(1)))                                         TWOH
      I2K=IABS(SLASHO(2,IV(2)))                                         TWOH
      DO 3 JC=1,LIM2                                                    TWOH
      DO 6 JD=1,4                                                       TWOH
      IF(I1B.NE.INT2(IX(1,JD),JC))GOTO6                                 TWOH
      IF(I1K.NE.INT2(IX(2,JD),JC))GOTO6                                 TWOH
      IF(I2B.NE.INT2(IX(3,JD),JC))GOTO6                                 TWOH
      IF(I2K.EQ.INT2(IX(4,JD),JC))GOTO7                                 TWOH
    6 CONTINUE                                                          TWOH
      GOTO3                                                             TWOH
    7 DO 4 JD=1,I1                                                      TWOH
    4 FAC2(JD,JC)=FAC2(JD,JC)+FAC(JD)*SIGN                              TWOH
      GO TO 2                                                           TWOH
    3 CONTINUE                                                          TWOH
      LIM2=LIM2+1                                                       TWOH
      IF(100.LT.LIM2)GOTO10                                             TWOH
      INT2(1,LIM2)=I1B                                                  TWOH
      INT2(2,LIM2)=I1K                                                  TWOH
      INT2(3,LIM2)=I2B                                                  TWOH
      INT2(4,LIM2)=I2K                                                  TWOH
      DO 5 JD=1,I1                                                      TWOH
    5 FAC2(JD,LIM2)=FAC(JD)*SIGN                                        TWOH
    2 CONTINUE                                                          TWOH
    1 CONTINUE                                                          TWOH
      RETURN                                                            TWOH
   10 WRITE(8,900)                                                      TWOH
  900 FORMAT('0',131('*')/20X,'MORE THAN 100 TWO-EL INTEGRALS'/131('*'))TWOH
      STOP                                                              TWOH
      END                                                               THRH
      SUBROUTINE THREE(I,J,K,L3,I1)                                     THRH
      IMPLICIT REAL*8 (A-H,O-Z)                                         THRH
      COMMON/FINT/LVEC(3,52),SLATOR(52,10                               THRH
      COMMON/SUBINT/FAC(3),FAC1(3,50),FAC2(3,100),FAC3(3,200),FAC4(3,300THRH
     .),INT1(50),INT2(4,100),INT3(6,200),INT4(8,300),SLASHO(2,10),DIFORBTHRH
     .(4)                                                               THRH
      INTEGER SLATOR,SLASHO,DIFORB,IV(3)                                THRH
      REAL*8 LVEC                                                       THRH
      LOGICAL SPIN                                                      THRH
      INTEGER IX(6,12)/1,2,3,4,5,6,3,4,1,2,5,6,5,6,3,4,1,2,1,2,5,6,3,4,5THRH
     .,6,1,2,3,4,3,4,5,6,1,2,2,1,4,3,6,5,4,3,2,1,6,5,6,5,4,3,2,1,2,1,6,5THRH
     .,4,3,6,5,2,1,4,3,4,3,6,5,2,1/                                     THRH
      SPIN(I,J,K,L,M,N)= (0.GT.ISIGN(1,I)*ISIGN(1,J)).OR.(0.GT.SIGN(1,KTHRH
     1)*ISIGN(1,L)).OR.(0.GT.ISIGN(1,M)*ISIGN(1,N))                     THRH
      I1B=IABS(SLASHO(1,I))
```

```
      I2B=IABS(SLASHO(1,J))                                              THRH
      I3B=IABS(SLASHO(1,K))                                              THRH
      DO 2 JA=1,3                                                        THRH
      DO 3 JB=1,3                                                        THRH
      IF(JA.EQ.JB) GO TO 3                                               THRH
      DO 4 JC=1,3                                                        THRH
      IF((JC.EQ.JA).OR.(JC.EQ.JB)) GO TO 4                               THRH
      IV(JA)=I                                                           THRH
      IV(JB)=J                                                           THRH
      IV(JC)=K                                                           THRH
      IF (SPIN(SLASHO(1,I),SLASHO(2,IV(1)),SLASHO(1,J),SLASHO(2,IV(2)),STHRH
     .LASHO(1,K),SLASHO(2,IV(3)))) GO TO 4                               THRH
      SIGN=SIG(3,JA,JB,JC,4)                                             THRH
      I1K=IABS(SLASHO(2,IV(1)))                                          THRH
      I2K=IABS(SLASHO(2,IV(2)))                                          THRH
      I3K=IABS(SLASHO(2,IV(3)))                                          THRH
      DO 5 JD=1,L3                                                       THRH
      DO 8 JE=1,12                                                       THRH
      IF(I1B.NE.INT3(IX(1,JE),JD))GOTO8                                  THRH
      IF(I1K.NE.INT3(IX(2,JE),JD))GOTO8                                  THRH
      IF(I2B.NE.INT3(IX(3,JE),JD))GOTO8                                  THRH
      IF(I2K.NE.INT3(IX(4,JE),JD))GOTO8                                  THRH
      IF(I3B.NE.INT3(IX(5,JE),JD))GOTO8                                  THRH
      IF(I3K.EQ.INT3(IX(6,JE),JD))GOTO9                                  THRH
      CONTINUE                                                           THRH
      GOTO5                                                              THRH
      DO 6 JE=1,I1                                                       THRH
    6 FAC3(JE,JD)=FAC3(JE,JD)+FAC(JE)*SIGN                               THRH
      GO TO 4                                                            THRH
    5 CONTINUE                                                           THRH
      L3=L3+1                                                            THRH
      IF(200.LT.L3)GOTO10                                                THRH
      INT3(1,L3) = I1B                                                   THRH
      INT3(2,L3) = I1K                                                   THRH
      INT3(3,L3) = I2B                                                   THRH
      INT3(4,L3)=I2K                                                     THRH
      INT3(5,L3)=I3B                                                     THRH
      INT3(6,L3)=I3K                                                     THRH
      DO 7 JE=1,I1                                                       THRH
    7 FAC3(JE,L3)=FAC(JE)*SIGN                                           THRH
    4 CONTINUE                                                           THRH
    3 CONTINUE                                                           THRH
    2 CONTINUE                                                           THRH
      RETURN                                                             THRH
   10 WRITE(8,900)                                                       THRH
      STOP                                                               THRH
  900 FORMAT('0',131('*')/30X,'MORE THAN 200 THREE*EL INTEGRALS'/131('*  THRH
     .'))                                                                THRH
      END                                                                THRH
      SUBROUTINE FOUR(I,J,K,L,L4,I1)                                     FORH
      IMPLICIT REAL*8 (A-H,O-Z)                                          FORH
      COMMON/FINT/LVEC(3,52),SLATOR(52,10)                               FORH
      COMMON/SUBINT/FAC(3),FAC1(3,50),FAC2(3,100),FAC3(3,200),FAC4(3,300FORH
     .),INT1(50),INT2(4,100),INT3(6,200),INT4(8,300),SLASHO(2,10),DIFORBFORH
     .(4)                                                                FORH
      INTEGER SLATOR,SLASHO,DIFORB,IV(4)                                 FORH
      REAL*8 LVEC                                                        FORH
      INTEGER IX(8,48)/1,2,3,4,5,6,7,8,3,4,1,2,5,6,7,8,5,6,3,4,1,2,7,8,7FORH
     .,8,3,4,5,6,1,2,1,2,5,6,3,4,7,8,1,2,7,8,5,6,3,4,1,2,3,4,7,8,5,6,5,6FORH
     .,1,2,3,4,7,8,3,4,5,6,1,2,7,8,7,8,1,2,5,6,3,4,3,4,7,8,5,6,1,2,7,8,3FORH
```

```
.,4,1,2,5,6,5,6,3,4,7,8,1,2,1,2,7,8,3,4,5,6,1,2,5,6,7,8,3,4,7,8,1,2FORH
.,3,4,5,6,7,8,5,6,1,2,3,4,5,6,1,2,7,8,3,4,3,4,7,8,1,2,5,6,5,6,7,8,3FORH
.,4,1,2,3,4,5,6,7,8,1,2,3,4,1,2,7,8,5,6,5,6,7,8,1,2,3,4,7,8,5,6,3,4FORH
.,1,2,2,1,4,3,6,5,8,7,4,3,2,1,6,5,8,7,6,5,4,3,2,1,8,7,8,7,4,3,6,5,2FORH
.,1,2,1,6,5,4,3,8,7,2,1,8,7,6,5,4,3,2,1,4,3,8,7,6,5,6,5,2,1,4,3,8,7FORH
.,4,3,6,5,2,1,8,7,8,7,2,1,6,5,4,3,4,3,8,7,6,5,2,1,8,7,4,3,2,1,6,5,6FORH
.,5,4,3,8,7,2,1,2,1,8,7,4,3,6,5,2,1,6,5,8,7,4,3,8,7,2,1,4,3,6,5,8,7FORH
.,6,5,2,1,4,3,6,5,2,1,8,7,4,3,4,3,8,7,2,1,6,5,6,5,8,7,4,3,2,1,4,3,6FORH
.,5,8,7,2,1,4,3,2,1,8,7,6,5,6,5,8,7,2,1,4,3,8,7,6,5,4,3,2,1/      FORH
 LOGICAL SPIN                                                        FORH
 SPIN(IB,IK,JZ,JK,KB,KK,LB,LK)= (0.GT.ISIGN(1,IB)*ISIGN(1,IK)).OR. FORH
1(0.GT.ISIGN(1,Z)*ISIGN(1,JK)).OR.(0.GT.ISIGN(1,KB)*ISIGN(1,KK)).OFORH
2R.(0.GT.ISIGN(1,LB)*ISIGN(1,LK))                                   FORH
 I1B=IABS(SLASHO(1,I))                                              FORH
 I2B=IABS(SLASHO(1,J))                                              FORH
 I3B=IABS(SLASHO(1,K))                                              FORH
 I4B=IABS(SLASHO(1,L))                                              FORH
 DO 1 JA=1,4                                                        FORH
 DO 2 JB=1,4                                                        FORH
 IF(JA.EQ.JB) GO TO 2                                               FORH
 DO 3 JC=1,4                                                        FORH
 IF((JC.EQ.JA).OR.(JC.EQ.JB)) GO TO 3                               FORH
 DO 4 JD=1,4                                                        FORH
IF((JD.Q.JA).OR.(JD.EQ.JB).OR.(JD.EQ.JC)) GO TO 4                 FORH
 IV(JA)=I                                                           FORH
 IV(JB)=J                                                           FORH
 IV(JC)=K                                                           FORH
 IV(JD)=L                                                           FORH
 IF(SPIN(SLASHO(1,I),SLASHO(2,IV(1)),SLASHO(1,J),SLASHO(2,IV(2)),SLFORH
.ASHO(1,K),SLASHO(2,IV(3)),SLASHO(1,L),SLASHO(2,IV(4)))) GO TO 4   FORH
 SIGN=SIG(4,JA,JB,JC,JD)                                            FORH
 I1K=IABS(SLASHO(2,IV(1)))                                          FORH
 I2K=IABS(SLASHO(2,IV(2)))                                          FORH
 I3K=IABS(SLASHO(2,IV(3)))                                          FORH
 I4K=IABS(SLASHO(2,IV(4)))                                          FORH
 DO 5 JE=1,L4                                                       FORH
 DO 8 JF=1,48                                                       FORH
 IF(I1B.NE.INT4(IX(1,JF),JE))GOTO8                                  FORH
 IF(I1K.NE.INT4(IX(2,JF),JE))GOTO8                                  FORH
 IF(I2B.NE.INT4(IX(3,JF),JE))GOTO8                                  FORH
 IF(I2K.NE.INT4(IX(4,JF),JE))GOTO8                                  FORH
 IF(I3B.NE.INT4(IX(5,JF),JE))GOTO8                                  FORH
 IF(I3K.NE.INT4(IX(6,JF),JE))GOTO8                                  FORH
 IF(I4B.NE.INT4(IX(7,JF),JE))GOTO8                                  FORH
 IF(I4K.EQ.INT4(IX(8,JF),JE))GOTO9                                  FORH
 CONTINUE                                                           FORH
 GOTO5                                                              FORH
 DO 6 JF=1,I1                                                       FORH
6 FAC4(JF,JE)=FAC4(JF,JE)+FAC(JF)*SIGN                              FORH
 GO TO 4                                                            FORH
5 CONTINUE                                                          FORH
 L4=L4+1                                                            FORH
 IF(300.LT.L4) GO TO 10                                             FORH
 INT4(1,L4)=I1B                                                     FORH
 INT4(2,L4)=I1K                                                     FORH
 INT4(3,L4)=I2B                                                     FORH
 INT4(4,L4)=I2K                                                     FORH
 INT4(5,L4)=I3B                                                     FORH
 INT4(6,L4)=I3K                                                     FORH
 INT4(7,L4)=I4B
```

```
      INT4(8,L4)=I4K                                                    FORH
      DO 7 JF=1,I1                                                      FORH
    7 FAC4(JF,L4)=FAC(JF)*SIGN                                          FORH
    4 CONTINUE                                                          FORH
    3 CONTINUE                                                          FORH
    2 CONTINUE                                                          FORH
    1 CONTINUE                                                          FORH
      RETURN                                                            FORH
   10 WRITE(8,900)                                                      FORH
  900 FORMAT('0',131('*')/20X,'MORE THAN 300 FOUR-EL-INTEGRALS'/131('*')FORH
     1)                                                                 FORH
      STOP                                                              FORH
      END                                                               FORH
      FUNCTION SIG(N,I,J,K,L)                                           FSIG
      REAL*8 SIG                                                        FSIG
      INTEGER IV(4)                                                     FSIG
      IV(1)=I                                                           FSIG
      IV(2)=J                                                           FSIG
      IV(3)=K                                                           FSIG
      IV(4)=L                                                           FSIG
      ISUM=0                                                            FSIG
      NM1=N-1                                                           FSIG
      DO 1 JA=1,NM1                                                     FSIG
      JA1=JA+1                                                          FSIG
      DO 1 JB=JA1,N                                                     FSIG
    1 IF(IV(JA).GT.IV(JB)) ISUM=ISUM+1                                  FSIG
      SIG=1.D0*DFLOAT((-1)**ISUM)                                       FSIG
      RETURN                                                            FSIG
      END                                                               FSIG
      REAL*8 LSSQMA(52,52)/2704*0.D0/,EIGVAL(52,52),EIGVEC(52,52),      LSQT
     1B(52),F1,F2,SPIN(22)                                              LSQT
      EQUIVALENCE(LSSQMA(1),EIGVAL(1))                                  LSQT
                                                                        LSQT
   THIS ROUTINE IS SET UP TO CALCULATE LS-EIGENFUNCTIONS OF UP          LSQT
   TO TEN ELECTRONS. TO INCLUDE A LARGER NUMBER OF ELECTRONS            LSQT
   SEEMS SENSELESS, SINCE RUSSELL-SAUNDERS COUPLING BREAKS DOWN         LSQT
                                                                        LSQT
   THE EIGENVALUES ARE ON THE AVERAGE ACCURATE TO 11 SIGNIFICANT        LSQT
   FIGURES. IF HIGHER ACCURACY IS DESIRED, CHANGE STATEMENT 5 IN        LSQT
   SUBROUTINE 'DEIGE'.                                                  LSQT
                                                                        LSQT
                                                                        LSQT
   THE ROUTINE CAN HANDLE STATES WHICH ARE REPRESENTED BY UP TO         LSQT
   52 SLATERDETERMINANTS. IF A LARGER NUMBER OF SLATORS ARISE, THE      LSQT
   FOLLOWING CHANGES HAVE TO BE MADE                                    LSQT
      CHANGE THE DIMENSIONS OF LSSSQMA,EIGVAL,EIGVEC,B,SLDV,NUMDET      LSQT
      IN THE MAIN PROGRAM.                                              LSQT
      CHANGE THE FORMAT STATEMENTS IN THE SUBROUTINE SHREIB             LSQT
      CHANGE DIMENSION OF SLDV,NUMDET IN SUBROUTINES                    LSQT
        DETVAR                                                          LSQT
        OPERAT                                                          LSQT
        LSSQUA                                                          LSQT
        COMP                                                            LSQT
        OUTPU                                                           LSQT
                                                                        LSQT
      INTEGER*2  DMAT(4,100), CONFIG ( 33), STATE (2), IVEC (20),       LSQT
     1ICOMV (20), ISTA (20), SLDV ( 52,4,20),NUMDET(52,20),CMAT(4,20)   LSQT
     3,LINE(22),STTE(2)                                                 LSQT
 INPUT IS AS FOLLOWS:                                                   LSQT
   M   THE NUMBER OF UNEQUIVALENT STATES TIMES 3                        LSQT
```

```
C   N   THE NUMBER OF ELECRONS                                          LSQT
C CONFIGURATION: 1S1 2P2 3P1   =  01 00 01 02 01 02 03 02 01           LSQT
C STATE          3P            =  03 01                                LSQT
C   PUT  M,N   CONF,STATE,   AS CONTINUOUS I4 INPUT                     LSQT
C IVEC CONTAINS THE POSITION FOR SLD IN DMAT                            LSQT
C ICOMPV CONTAINS THE MAXIMUM IVEC CAN REACH                            LSQT
C                                                                       LSQT
C   AFTER THE STATE WITH ML=L AND MS=S HAS BEEN COMPUTED, - & S- ARE   LSQT
C   APPLIED(IN SUBROUTINE LMINUS) TO OBTAIN ALL POSSIBLE E'FNS FOR ALL  LSQT
C   VALUES OF ML AND MS.                                                LSQT
C                                                                       LSQT
C   THE OUTPUT IS WRITTEN ON UNIT(6)                                    LSQT
C                                                                       LSQT
C   INTCOE COMPUTES THE INTEGRALS OBTAINED BY OPERATING WITH ONE-,      LSQT
C   TWO-,THREE-,AND FOUR-ELECTRON-OPERATORS                             LSQT
C   IT WRITES THE RESULTS ON UNIT(8)                                    LSQT
C                                                                       LSQT
    1 READ(5,901,END=23)M,N,(CONFIG(J1),J1=1,M),(STATE(J1),J1=1,2)      LSQT
      DO 2 J1 = 1,52                                                    LSQT
      DO 2 J2 = 1,52                                                    LSQT
    2 LSSQMA(J1,J2) = 0.D0                                              LSQT
      CALL VECT(IVEC,ISTA,ICOMV,M,CONFIG,N)                             LSQT
      CALL EXPAND (STATE,CONFIG,DMAT,N,M)                               LSQT
C AFTER DMAT IS COMPUTED IT'S COLUMNS ARE USED TO SET UP ALL POSSIBLE   LSQT
C SLATERDETERMINANTS, WHICH ARE CHECKED IF THEY FULFILL THE STATE COND. LSQT
      K=0                                                               LSQT
      I1=N                                                              LSQT
    9 CALL DETVAR (DMAT,IVEC,SLDV,N,STATE,K,NUMDET,&45)                 LSQT
      IVEC(I1) = IVEC (I1) + 1                                          LSQT
      IF (IVEC (I1) .LT. ICOMV(I1)) GO TO 9                             LSQT
      CALL RESET (IVEC,I1,&19,&29,N,ISTA,&19)                           LSQT
   19 CALL CHECK (IVEC,ISTA,&9 ,N ,&29,ICOMV)                           LSQT
   29 STTE(1) = STATE(1)                                                LSQT
      STTE(2) = STATE(2)                                                LSQT
      CALL OUTPU (SLDV,STATE,CONFIG,K,N,&45,0,STTE)                     LSQT
      CALL OPERAT (SLDV,LSSQMA,K,STATE,CMAT,N,EIGVEC,3,&45,I1)          LSQT
      CALL LMINUS (SLDV,EIGVEC,I1,K,N,CONFIG,STATE,M,EIGVAL)            LSQT
      CALL INTCOE(N,I1,STATE)                                           LSQT
   45 GO TO 1                                                           LSQT
  901 FORMAT (20 I 4)                                                   LSQT
   23 STOP                                                              LSQT
      END                                                               LSQT
      SUBROUTINE LMINUS (SLDV1,MAT  ,I1,K,N,CONFIG,STATE,M,EIVEC)       LMIN
C THIS ROUTINE GENERATES ALL THE POSSIBLE FUNCTIONS WITH A GIVEN L AND  LMIN
C S-VALUE. FIRST L-MINUS IS APPLIED, THEN S-MINUS, AND THEN THE SLATER- LMIN
C DETERMINANTS AND EIGENVECTORS ARE PRINTED OUT.                        LMIN
C                                                                       LMIN
      INTEGER*2  SLDV1(52,4,10),SLDV2(52,4,10),CONFIG(33),STATE(2)      LMIN
     1,STTE(2)                                                          LMIN
      INTEGER  IVE(10)                                                  LMIN
      REAL*8 EIVEC (52,52),F,FAC(10),MAT(K,K)                           LMIN
C                                                                       LMIN
C  THE FOLLOWING STMTS CHECK IF A CLOSED SHELL IS PRESENT.IF SO,IT WILL LMIN
C  BE DISREGARDED FOR THE OPERATION OF L-MINUS OR S-MINUS               LMIN
C                                                                       LMIN
      DO 110 J110 = 1,I1                                                LMIN
      DO 110 J111 = 1,K                                                 LMIN
  110 EIVEC(J111,J110) = MAT(J111,K+1-J110)                             LMIN
      ICFILL = 1                                                        LMIN
      CALL FILL (SLDV1,EIVEC,K,ICFILL,N,-1,I1)                          LMIN
```

```
      ICOUNT=0
      ILIM1=1                                                           LMIN
      DO 10 J1 = 2,M,3                                                  LMIN
      IF ((CONFIG(J1)*2+1)*2.EQ. (CONFIG(J1+1)))  GO TO 11              LMIN
      ILIM2 =ILIM1+ CONFIG(J1+1) - 1                                    LMIN
      DO 101 J11 =ILIM1,ILIM2                                           LMIN
      ICOUNT=ICOUNT+1                                                   LMIN
  101 IVE(J11)= ICOUNT                                                  LMIN
      ILIM1 = ILIM2 +1                                                  LMIN
      GO TO 10                                                          LMIN
   11 ICOUNT = ICOUNT +CONFIG (J1+1)                                    LMIN
   10 CONTINUE                                                          LMIN
C                                                                       LMIN
C    IVE CONTAINS NOW ALL ORBITALS TO BE OPERATED UPON                  LMIN
C    ILIM2 SPECIFIES THE NUMBER OF THESE ORBITALS                       LMIN
C                                                                       LMIN
C    K2 CONTAINS THE NUMBER OF SLATORS IN SLDV2                         LMIN
C    ISIND IS THE INDICATOR TO SHOW IF ONE HAS TO OPERATE WITH          LMIN
C    S-PLUS OR S-MINUS                                                  LMIN
      STTE(1) = STATE(1)                                                LMIN
      STTE(2) = STATE(2)                                                LMIN
      IS2 = 1                                                           LMIN
      ISIND = 1                                                         LMIN
C                                                                       LMIN
C    STATE(1) IS EQUAL TO 2S+1                                          LMIN
C    STATE(2) IS EQUAL TO L                                             LMIN
C    STTE IS A VARIABLE WHICH CONTAINS MS AND ML. SINCE THE ORIGINAL    LMIN
C    EIGENF'NS OF L AND S OP. ARE COMPUTED FOR THE HIGHEST MS AND       LMIN
C    ML VALUES, STTE IS IDENTICAL TO STATE AT THE START OF THE ROUTINE  LMIN
C                                                                       LMIN
C                                                                       LMIN
C    THE FOLLOWING METHOD IS USED TO OPERATE UPON F(L,S,ML,MS) WITH     LMIN
C    S- AND L-..                                                        LMIN
C      THE VARIABLE MAT CONTAINS THE ORIGINAL E'VECTORS. THE FIRST      LMIN
C      K COLUMNS AND K ROWS OF EIVEC ARE FILLED WITH MAT.               LMIN
C      THE VARIABLE ISIND IS SET TO +1 TO INDICATE THAT THE E'VECTOR    LMIN
C      TO BE OPERATED UPON ARE IN THE FIRST I1 COLUMNS, THE E'VECTORS   LMIN
C      OBTAINED BY OPERATION ARE TO BE PUT IN THE LAST I1 COLUMNS       LMIN
C      OF THE (52,52) MATRIX EIVEC.                                     LMIN
C                                                                       LMIN
C    IF WE HAVE A SINGELET-STATE, NO OPERATION WITH S- OR S+            LMIN
    9 IF (STATE(1) .EQ. 1) GO TO 18                                     LMIN
      IF (ISIND) 13,13,12                                               LMIN
   12 CALL SOP(SLDV1,SLDV2,K,K2,EIVEC,FAC,ISIND,IVE,ILIM2,CONFIG,STATE, LMIN
     1STTE,IS2,N,I1,ICFILL)                                             LMIN
      GO TO 14                                                          LMIN
   13 CALL SOP(SLDV2,SLDV1,K2,K,EIVEC,FAC,ISIND,IVE,ILIM2,CONFIG,STATE, LMIN
     1STTE,IS2,N,I1,ICFILL)                                             LMIN
   14 ISIND = ISIND*(-1)                                                LMIN
      IF((STTE(1) .GT. 1) .AND. (STTE(1) .LT. STATE(1)))GO TO 9         LMIN
C    IF WE HAVE A S-STATE,I.E. STATE(2)=0 NO OPERATION WITH L-          LMIN
   18 IF (STATE(2) .EQ. 0) RETURN                                       LMIN
      IF (STTE(2)*(-1) .GE. STATE(2)) RETURN                            LMIN
      IF (ISIND) 15,15,16                                               LMIN
   15 CALL LOP (SLDV2,SLDV1,K2,K,EIVEC,FAC,IVE,ILIM2,N,I1,ISIND,        LMIN
     1STATE,CONFIG,STTE,ICFILL)                                         LMIN
      GO TO 17                                                          LMIN
   16 CALL LOP(SLDV1,SLDV2,K,K2,EIVEC,FAC,IVE,ILIM2,N,I1,ISIND,         LMIN
     1STATE,CONFIG,STTE,ICFILL)                                         LMIN
   17 IS2 = IS2*(-1)                                                    LMIN
```

```
      ISIND = ISIND * (-1)                                              LMIN
      GO TO 9                                                           LMIN
      END                                                               LMIN
      SUBROUTINE LOP (SL1,SL2,K1,K2,EIVEC,FAC,IVE,ILIM2,N,I1,ISIND,     L-OP
     1STATE,CONFIG,STTE,ICFILL)                                         L-OP
C THIS ROUTINE OPERATES WITH L-MIN ON SL1, STORES THE RESULT IN SL2     L-OP
      REAL*8 EIVEC(52,52),FAC(10),F,DSQRT,DFLOAT                        L-OP
      INTEGER*2 SL1(52,4,10),SL2(52,4,10),SL3(1,4,10)                   L-OP
     1,STATE(2),CONFIG(33),STTE(2)                                      L-OP
      INTEGER IVE (10)                                                  L-OP
C                                                                       L-OP
C DO LOOP 10 RUNS THRU ALL SLATORS                                      L-OP
C LOOP 11 THRU ALL COLUMNS WHICH DO NOT BELONG TO CLOSED SHELLS         L-OP
      DO 30 J30 = 1,52                                                  L-OP
      DO 30 J31 = 1,4                                                   L-OP
      DO 30 J32 = 1,10                                                  L-OP
   30 SL2(J30,J31,J32) = 0                                              L-OP
      I2 = 0                                                            L-OP
      IF (ISIND .EQ. -1) I2 = 52 - I1                                   L-OP
      K2 = 1                                                            L-OP
      DO 10  J1 =1,K1                                                   L-OP
      DO 11  J11=1,ILIM2                                                L-OP
C                                                                       L-OP
C THE NEXT STMT CHECKS IF ABS(ML).LT.(L)                                L-OP
C                                                                       L-OP
      IF (SL1(J1,2,IVE(J11)).LE.(-1)* (SL1(J1,3,IVE(J11))))GO TO 11     L-OP
C                                                                       L-OP
C IF SO, SL1(J1,*,*) IS PUT INTO SL3(1,*,*)                             L-OP
C                                                                       L-OP
      DO 12 J12 =1,N                                                    L-OP
      DO 12 J121=1,4                                                    L-OP
   12 SL3( 1,J121,J12) = SL1(J1,J121,J12)                               L-OP
C                                                                       L-OP
C THE FACTOR FOR L-MINUS IS CALCULATED                                  L-OP
C                                                                       L-OP
      LMU = SL3(1,2,IVE(J11))                                           L-OP
      MLMU = SL3(1,3,IVE(J11))                                          L-OP
      F=DSQRT(DFLOAT(LMU*(LMU+1)-MLMU*(MLMU-1)))                        L-OP
      DO 13 J13 =1,I1                                                   L-OP
   13 FAC(J13)= EIVEC(J1,I2+J13)*F                                      L-OP
C                                                                       L-OP
C THE ML-VALUE IS DECREASED BY ONE                                      L-OP
C                                                                       L-OP
      SL3(1,3,IVE(J11)) = SL3( 1,3,IVE(J11))-1                          L-OP
C                                                                       L-OP
C   LOOP 40 CHECKS IF BY OPERATION WITH L- TWO ORBITALS HAVE            L-OP
C   BECOME IDENTICAL,THUS VIOLATING THE PAULI PRINCIPLE.                L-OP
C                                                                       L-OP
      DO 40 J40 = 1,ILIM2                                               L-OP
      IF(J40 .EQ. J11) GO TO 40                                         L-OP
      IF (0 .EQ. J11+J42) GO TO 40                                      L-OP
      DO 41 J41 = 1,4                                                   L-OP
      IF(SL3(1,J41,IVE(J40)) .NE. SL3(1,J41,IVE(J11)))GO TO 40          L-OP
C                                                                       L-OP
C                                                                       L-OP
                                                                        L-OP
   41 CONTINUE                                                          L-OP
      GO TO 11                                                          L-OP
   40 CONTINUE                                                          L-OP
      CALL SEARCH (SL2,K2,EIVEC,FAC,I1,IVE,ILIM2,ISIND,SL3,N)           L-OP
C                                                                       L-OP
```

```
C SEARCH LOOKS IF SL3(1,*,*) IS ALREADY CONTAINED IN SL2                      L-OP
C                                                                             L-OP
 11  CONTINUE                                                                 L-OP
 10  CONTINUE                                                                 L-OP
     STTE(2) = STTE(2)-1                                                      L-OP
     K2 = K2 - 1                                                              L-OP
     CALL OUTPU (SL2,STATE,CONFIG,K2,N,&21,1,STTE)                            L-OP
 21  J22 = 52 -  I1                                                           L-OP
     IF (ISIND .NE. 1) J22 = 0                                                L-OP
C  LOOP 50 NORMALIZES THE EIGENVECTORS                                        L-OP
     DO 50 J50 =1,I1                                                          L-OP
     F = 0.D0                                                                 L-OP
     DO 51 J51 = 1,K2                                                         L-OP
 51  F = F + EIVEC(J51,J22+J50)**2                                            L-OP
     DO 52 J52 = 1,K2                                                         L-OP
 52  EIVEC(J52,J22+J50) = EIVEC(J52,J22+J50)/DSQRT(F)                         L-OP
 50  CONTINUE                                                                 L-OP
     DO 20 J2 = 1,I1                                                          L-OP
  20 WRITE(6,900) J2,(J21,EIVEC(J21,J22+J2),J21=1,K2)                         L-OP
     CALL FILL(SL2,EIVEC,K2,ICFILL,N,ISIND,I1)                                L-OP
 900 FORMAT (////'  EIGENVECTOR',I4,'.'//10(4(4X,I3,')',D25.15)/))            L-OP
     RETURN                                                                   L-OP
     END                                                                      L-OP
     SUBROUTINE SOP (SL1,SL2,K1,K2,EIVEC,FAC,ISIND,IVE,ILIM2,CONFIG,          S-OP
    1STATE,STTE,IS2,N,I1,ICFILL)                                              S-OP
     INTEGER*2 SL1(52,4,10),SL2(52,4,10) ,STATE(2),CONFIG(33),STTE(2)         S-OP
    1,SL3(1,4,10)                                                             S-OP
     INTEGER IVE(10)                                                          S-OP
C  THIS ROUT. OPERATES WITH S+ OR S- ON SL1, STORING THE RESULT IN SL2        S-OP
C  SL2 IS THEN PRINTED OUT TOGETHER WITH THE EIGENVECTORS                     S-OP
     REAL*8 EIVEC(52,52),FAC(10),F                                            S-OP
     DO 30 J30 = 1,52                                                         S-OP
     DO 30 J31 = 1,4                                                          S-OP
     DO 30 J32 = 1,10                                                         S-OP
 30  SL2(J30,J31,J32) = 0                                                     S-OP
     K2 = 1                                                                   S-OP
     DO 10 J1=1,K1                                                            S-OP
C                                                                             S-OP
C   IND IS A NUMBER WHICH IS SET TO -1 IF A SPACE-ORBITAL                     S-OP
C   IS REPRESENTED TWICE                                                      S-OP
C                                                                             S-OP
                                                                              S-OP
     IND = 1                                                                  S-OP
     DO 11 J11=1,ILIM2                                                        S-OP
     IF (IND) 12,12,13                                                        S-OP
 13  IF (J11.EQ.ILIM2)GO TO 14                                                S-OP
C                                                                             S-OP
C  FIRST IT IS COMPARED, IF TWO ADJACENT COLUMNS ARE EQUAL IN THE FIRST       S-OP
C  THREE QUANTUMNUMBERS. IF SO, J11 IS INCREASED BY TWO TIMES ONE             S-OP
C                                                                             S-OP
                                                                              S-OP
     DO 113 J113 = 1,3                                                        S-OP
     IF(SL1(J1,J113,IVE(J11)) .NE. SL1(J1,J113,IVE(J11+1)))GO TO 14           S-OP
 113 CONTINUE                                                                 S-OP
     IND=-1                                                                   S-OP
     GO TO 11                                                                 S-OP
C                                                                             S-OP
C  STMT 14 CHECKS IF ONE CAN OPERATE WITH S+ OR S- ON SL1                     S-OP
C                                                                             S-OP
 14  IF(SL1(J1,4,IVE(J11)).NE.IS2  )GO TO 11                                  S-OP
C                                                                             S-OP
C   LOOP 15 PUTS SL1(J1,*,*) INTO SL3(1,*,*)                                  S-OP
```

```
C                                                                       S-OP
      DO 15 J15=1,N                                                     S-OP
      DO 15 J151=1,4                                                    S-OP
 15   SL3( 1,J151,J15)=SL1(J1,J151,J15)                                 S-OP
C                                                                       S-OP
C   THIS STMT CHANGES SL3(1,4,*) INTO ITS NEGATIVE                      S-OP
C                                                                       S-OP
      SL3( 1,4,IVE(J11))=SL3 (1,4,IVE(J11))*(-1)                        S-OP
C                                                                       S-OP
C   LOOP 215 CHECKS IF TWO ORBITALS ARE IDENTICAL                       S-OP
C                                                                       S-OP
      DO 215 J215 = 1,ILIM2                                             S-OP
      IF(J11 .EQ. J215) GO TO 215                                       S-OP
      DO 216 J216 = 1,4                                                 S-OP
      IF(SL3(1,J216,IVE(J11)) .NE. SL3(1,J216,IVE(J215)))GO TO 215      S-OP
 216  CONTINUE                                                          S-OP
      GO TO 11                                                          S-OP
 215  CONTINUE                                                          S-OP
C  SEARCH LOOKS IF IDENTICAL SLATOR IS CONTAINED IN SI2                 S-OP
C                                                                       S-OP
      I2 = 0                                                            S-OP
      IF (ISIND .EQ. -1) I2=52-I1                                       S-OP
      DO 115 J115 = 1,I1                                                S-OP
  115 FAC(J115) = EIVEC(J1,I2+J115)                                     S-OP
      CALL SEARCH (SL2,K2,EIVEC,FAC,I1,IVE,ILIM2,ISIND,SL3,N)           S-OP
 12   IND=1                                                             S-OP
 11   CONTINUE                                                          S-OP
 10   CONTINUE                                                          S-OP
      K2 = K2 - 1                                                       S-OP
      STTE(1) = STTE(1) - IS2                                           S-OP
      CALL OUTPU (SL2,STATE,CONFIG,K2,N,&21,1,STTE)                     S-OP
 21   J22 = 52 -  I1                                                    S-OP
      IF (ISIND .NE. 1) J22 = 0                                         S-OP
C   LOOP 50 NORMALIZES THE EIGENVECTORS                                 S-OP
      DO 50 J50 =1,I1                                                   S-OP
      F = 0.D0                                                          S-OP
      DO 51 J51 = 1,K2                                                  S-OP
 51   F = F + EIVEC(J51,J22+J50)**2                                     S-OP
      DO 52 J52 = 1,K2                                                  S-OP
 52   EIVEC(J52,J22+J50) = EIVEC(J52,J22+J50)/ DSQRT(F)                 S-OP
 50   CONTINUE                                                          S-OP
      DO 20 J2 = 1,I1                                                   S-OP
   20 WRITE(6,900) J2,(J21,EIVEC(J21,J22+J2),J21=1,K2)                  S-OP
      CALL FILL(SL2,EIVEC,K2,ICFILL,N,ISIND,I1)                         S-OP
  900 FORMAT (////'  EIGENVECTOR',I4,'.'//10(4(4X,I3,')',D25.15)/))     S-OP
      RETURN                                                            S-OP
      END                                                               S-OP
      SUBROUTINE SEARCH (SL,K,EIVEC,FAC,I1,IVE,ILIM2,ISIND,SP,N)        SRCH
C                                                                       SRCH
C  SEARCH LOOKS IF SL(K,*,*) IS CONTAINED IN SI(1-K-1,*,*)              SRCH
C                                                                       SRCH
      REAL*8 EIVEC(52,52),FAC(10),DFLOAT                                SRCH
      INTEGER*2 SL(52,4,10)  ,SP(1,4,10)                                SRCH
      INTEGER IVE(10)                                                   SRCH
C                                                                       SRCH
C  IF SL(K,*,*) IS EQUAL SL(J1,*,*),THEN ONLY THE ENTRY IN EIVEC IS     SRCH
C  CHANGED                                                              SRCH
C                                                                       SRCH
      I2=52-I1                                                          SRCH
      IF(ISIND.EQ.-1) I2=0                                              SRCH
```

```
      KM1=K-1
      DO 10 J1=1,KM1                                                    SRCH
      ISIGN = 1                                                         SRCH
      DO 11 J11 =1,ILIM2                                                SRCH
      DO 12 J12 =1,ILIM2                                                SRCH
      DO 13 J13 =1,4                                                    SRCH
      IF (SP(1,J13,IVE(J11)) .NE. SL(J1,J13,IVE(J12))) GO TO 12         SRCH
   13 CONTINUE                                                          SRCH
      IF (J11.NE.J12) ISIGN=ISIGN + 1                                   SRCH
      GO TO 11                                                          SRCH
   12 CONTINUE                                                          SRCH
      GO TO 10                                                          SRCH
   11 CONTINUE                                                          SRCH
      IF (ISIGN .EQ.1) ISIGN =2                                         SRCH
      DO 15  J15=1,I1                                                   SRCH
   15 EIVEC (J1,I2+J15)= EIVEC (J1,I2+J15)+FAC(J15)*DFLOAT((-1)**ISIGN) SRCH
      RETURN                                                            SRCH
   10 CONTINUE                                                          SRCH
      DO 16 J16 =1,I1                                                   SRCH
   16 EIVEC(K,I2+J16)= FAC(J16)                                         SRCH
      DO 20 J20 = 1,4                                                   SRCH
      DO 20 J21 = 1,N                                                   SRCH
   20 SL(K,J20,J21) = SP(1,J20,J21)                                     SRCH
      K=K+1                                                             SRCH
      IF (K.LT.53) RETURN                                               SRCH
      WRITE (6,900)                                                     SRCH
  900 FORMAT ('0 THERE ARE MORE THAN 53 SLATORS')                       SRCH
      STOP                                                              SRCH
      END                                                               SRCH
      SUBROUTINE FILL (SL,EIGVEC,K,ICFILL ,NOE,ISIND,I1)                FILT
C                                                                       FILT
C    THIS ROUTINE COMPRESSES THE SLATORS FROM 4-ON TO 1-ON              FILT
C    AND WRITES THE RESULT AS SLATOR(ICFILL,K,NOE) ON UNIT 3            FILT
C    THE EIGVECS ARE WRITTEN ON UNIT 4                                  FILT
C    THE FIRST RECORD OF EACH WRITE CONTAINS THE NO. OF SLATORS  AND    FILT
C    TNE NO. OF EIGVECS                                                 FILT
C                                                                       FILT
C                                                                       FILT
C    SLATOR(*,*,*) CONTAINS THE COMPRESED INDEX CALCULATED              FILT
C    FROM SL(*,*,*)                                                     FILT
C    N                                                                  FILT
C    L                                                                  FILT
C    ML  ARE SELFEXPLANATORY                                            FILT
C    MS                                                                 FILT
C                                                                       FILT
      REAL*8 EIGVEC(52,52),LVEC                                         FILT
      COMMON/FILIN/LVEC(20,3,52),SLATOR,KVEC(20)                        FILT
      INTEGER*2 SL(52,4,10),SLATOR(20,52,10)                            FILT
      INDG(N) = (N-2)*9-3                                               FILT
      INDF(L,M) = L+L*L+M                                               FILT
      KVEC(ICFILL)=K                                                    FILT
      DO 10 J10 = 1,K                                                   FILT
      DO 10 J11 = 1,NOE                                                 FILT
      N = SL(J10,1,J11)                                                 FILT
      L = SL(J10,2,J11)                                                 FILT
      ML= SL(J10,3,J11)                                                 FILT
      MS= SL(J10,4,J11)                                                 FILT
      LML = INDF (L,ML)                                                 FILT
      IF (N .LE. 2) GO TO 1                                             FILT
      INCOMP = (LML+INDG(N))*MS                                         FILT
```

```
      GO TO 10                                                          FILT
    1 INCOMP = (LML+N)*MS                                               FILT
   10 SLATOR (ICFILL,J10,J11) = INCOMP                                  FILT
      J22 = 0                                                           FILT
      IF (ISIND .EQ. 1) J22 = 52 - I1                                   FILT
      DO 20 JA=1,K                                                      FILT
      DO 20 JB=1,I1                                                     FILT
20    LVEC(ICFILL,JB,JA)=EIGVEC(JA,J22+JB)                              FILT
      ICFILL = ICFILL + 1                                               FILT
  900 FORMAT (20I4)                                                     FILT
  901 FORMAT (3D26.18)                                                  FILT
  902 FORMAT (2I4)                                                      FILT
      RETURN                                                            FILT
      END                                                               FILT
      SUBROUTINE INTCOE(NOE,I1,STATE)                                   INTT
C PURPOSE:                                                              INTT
C     TO COMPUTE SYMBOLICALLY THE INTEGRALS WHICH ARE OBTAINED WHEN     INTT
C     APPLYING THE OPERATOR H, HH AND SUMMING OVER A COMPLETE SET OF    INTT
C     L-S-EIGENSTATES                                                   INTT
C     VARIABLES:                                                        INTT
C     TERM: THE TERMSYMBOL, EQUIVALENT TO STATE IN 'LSO'                INTT
C     INT*: ARRAYS IN WHICH THE SYMBOLIC FORM OF THE INTEGRALS IS STOREDINTT
C     FAC*: ARRAYS IN WHICH THE COMPUTE COEFFICIENTS ARE STORED         INTT
      IMPLICIT REAL*8 (A-H,O-Z)                                         INTT
      COMMON/FILIN/LVEC(20,3,52),SLATOR,KVEC(20)                        INTT
      COMMON/ITC/FAC,RECODE,SLASHO,DIFORB                               INTT
      COMMON/ITC2/FAC1,FAC2,FAC3,FAC4,INT1,INT2,INT3,INT4,LIM1,LIM2,LIM3INTT
     .,LIM4                                                             INTT
      REAL*8 LVEC,FAC1(3,50),FAC2(3,100),FAC3(3,200),FAC4(3,300),FAC(3) INTT
      INTEGER INT1 (50),INT2(4,100),INT3(6,200),INT4(8,300),DIFORB(4),  INTT
     1RECODE,SLASHO(2,10)                                               INTT
      INTEGER*2 SLATOR(20,52,10),STATE(2)                               INTT
      LIM=STATE(1)*(STATE(2)*2+1)                                       INTT
      LIM1=0                                                            INTT
      LIM2=0                                                            INTT
      LIM3=0                                                            INTT
      LIM4=0                                                            INTT
      WRITE(8,910)                                                      INTT
      DO 12 JA=1,LIM                                                    INTT
      K=KVEC(JA)                                                        INTT
      WRITE(8,908) ((SLATOR(JA,IA1,IA2),IA2=1,NOE),IA1=1,K)             INTT
12    WRITE(8,909)((LVEC(JA,IA1,IA2),IA2=1,K),IA1=1,I1)                 INTT
      IF(I1.GT.3) GO TO 11                                              INTT
      DO 1 JA=1,LIM                                                     INTT
      K=KVEC(JA)                                                        INTT
      DO 1 JB=1,K                                                       INTT
      DO 1 JC=JB,K                                                      INTT
      CALL COMP1(JA,JB,JC,I1,&1,NOE)                                    INTT
      CALL SORT(NOE,I1)                                                 INTT
    1 CONTINUE                                                          INTT
      DLIM=DFLOAT(LIM)                                                  INTT
      DO 13 JA=1,I1                                                     INTT
      DO 14 JB=1,LIM1                                                   INTT
14    FAC1(JA,JB)=FAC1(JA,JB)/DLIM                                      INTT
      DO 15 JB=1,LIM2                                                   INTT
15    FAC2(JA,JB)=FAC2(JA,JB)/DLIM                                      INTT
      DO 16 JB=1,LIM3                                                   INTT
16    FAC3(JA,JB)=FAC3(JA,JB)/DLIM                                      INTT
      DO 17 JB=1,LIM4                                                   INTT
17    FAC4(JA,JB)=FAC4(JA,JB)/DLIM
```

```
13    CONTINUE
      WRITE(8,900)LIM1,LIM2,LIM3,LIM4                                   INTT
      DO 7 JA=1,LIM1                                                    INTT
    7 WRITE(8,903) INT1(JA),(FAC1(IA,JA),IA=1,I1)                       INTT
      DO 8 JA=1,LIM2                                                    INTT
    8 WRITE(8,904)(INT2(IA,JA),IA=1,4),(FAC2(IB,JA),IB=1,I1)            INTT
      IF(NOE.LT.3)RETURN                                                INTT
      DO 9 JA = 1,LIM3                                                  INTT
    9 WRITE(8,905) (INT3(IA,JA),IA=1,6),(FAC3(IA,JA),IA=1,I1)           INTT
      IF(NOE.LT.4)RETURN                                                INTT
      DO 10 JA=1,LIM4                                                   INTT
   10 WRITE(8,906) (INT4(IA,JA),IA=1,8),(FAC4(IA,JA),IA=1,I1)           INTT
      RETURN                                                            INTT
   11 WRITE(8,907)                                                      INTT
      STOP                                                              INTT
  900 FORMAT(20I4)                                                      INTT
  901 FORMAT(3D26.18)                                                   INTT
  902 FORMAT(2I4)                                                       INTT
  903 FORMAT(33X,I3,3D25.15)                                            INTT
  904 FORMAT(18X,2(3X,2I3),3D25.15)                                     INTT
  905 FORMAT(9X,3(3X,2I3),3D25.15)                                      INTT
  906 FORMAT(4(3X,2I3),3D25.15)                                         INTT
  907 FORMAT('0',131('*')/40X,'MORE THAN THREE LINEARLY INDEPENT EIGENFUINTT
     1NCTIONS'/131('*'))                                                INTT
908   FORMAT(' ',20I4)                                                  INTT
909   FORMAT(' ',10D12.4)                                               INTT
910   FORMAT('1')                                                       INTT
      END                                                               INTT
      SUBROUTINE COMP1(LI,I,J,I1,*,NOE)                                 COMT
      REAL*8 LVEC(20,3,52),FAC(3),FACT                                  COMT
      INTEGER SLASHO(2,10),RECODE,DIFORB(4)                             COMT
      INTEGER*2 SLATOR(20,52,10)                                        COMT
      COMMON/ITC/FAC,RECODE,SLASHO,DIFORB                               COMT
      COMMON/FILIN/LVEC,SLATOR,KVEC(20)                                 COMT
      ISUM=0                                                            COMT
      RECODE=0                                                          COMT
   11 DO 1 JA=1,NOE                                                     COMT
      SLASHO(1,JA)=SLATOR(LI,I,JA)                                      COMT
    1 SLASHO(2,JA)=SLATOR(LI,J,JA)                                      COMT
      FACT=1.D0                                                         COMT
      IF (J.EQ.I) GO TO 3                                               COMT
      FACT=2.D0                                                         COMT
      DO 5 JA=1,NOE                                                     COMT
      DO 6 JB=1,NOE                                                     COMT
      IF(SLASHO(1,JA).EQ.SLASHO(2,JB)) GO TO 2                          COMT
    6 CONTINUE                                                          COMT
      RECODE=RECODE+1                                                   COMT
      DIFORB(RECODE)=JA                                                 COMT
      IF (RECODE.GT.4) RETURN1                                          COMT
      GO TO 5                                                           COMT
    2 IF(JA.EQ.JB) GO TO 5                                              COMT
      ISUM = ISUM+1                                                     COMT
      IEX=SLASHO(2,JA)                                                  COMT
      SLASHO(2,JA) = SLASHO(2,JB)                                       COMT
      SLASHO(2,JB)=IEX                                                  COMT
    5 CONTINUE                                                          COMT
    3 DO 7 JA=1,I1                                                      COMT
    7 FAC(JA)=LVEC(LI,JA,I)*LVEC(LI,JA,J)*DFLOAT((-1)**ISUM)*FACT       COMT
      IF(RECODE.EQ.0) RECODE=1                                          COMT
      RETURN
```

```
      END                                                               COMT
      SUBROUTINE SORT(NOE,I1)                                           SRTT
      COMMON/ITC/FAC,RECODE,SLASHO,DIFORB                               SRTT
      COMMON/ITC2/FAC1,FAC2,FAC3,FAC4,INT1,INT2,INT3,INT4,LIM1,LIM2,LIM3SRTT
     .,LIM4                                                             SRTT
      REAL*8 FAC(3),FAC1(3,50),FAC2(3,100),FAC3(3,200),FAC4(3,300)      SRTT
      INTEGER SLASHO(2,10),DIFORB(4),INT1(50),INT2(4,100),INT3(6,200),  SRTT
     1INT4(8,300),RECODE                                                SRTT
      GO TO (1,2,3,4),RECODE                                            SRTT
    1 DO 7 JA=1,NOE                                                     SRTT
      CALL ONE (SLASHO,JA,FAC1,INT1,FAC,LIM1,I1)                        SRTT
      JBL=JA+1                                                          SRTT
      IF (JBL.GT.NOE)GOTO 7                                             SRTT
      DO 6 JB=JBL,NOE                                                   SRTT
      CALL TWO(SLASHO,JA,JB,FAC2,INT2,FAC,LIM2,I1)                      SRTT
      IF(NOE.LT.3) GO TO 6                                              SRTT
      JCL=JB+1                                                          SRTT
      IF(JCL.GT.NOE)GOTO 6                                              SRTT
      DO 5 JC=JCL,NOE                                                   SRTT
      CALL THREE(SLASHO,JA,JB,JC,FAC3,INT3,FAC,LIM3,I1)                 SRTT
      IF (NOE.LT.4) GO TO 5                                             SRTT
      JDL=JC+1                                                          SRTT
      IF(JDL.GT.NOE)GOTO 5                                              SRTT
      DO 8 JD=JDL,NOE                                                   SRTT
      CALL FOUR(SLASHO,JA,JB,JC,JD,FAC4,INT4,FAC,LIM4,I1)               SRTT
    8 CONTINUE                                                          SRTT
    5 CONTINUE                                                          SRTT
    6 CONTINUE                                                          SRTT
    7 CONTINUE                                                          SRTT
      RETURN                                                            SRTT
    2 JA=DIFORB(1)                                                      SRTT
      JB=DIFORB(2)                                                      SRTT
      CALL TWO(SLASHO,JA,JB,FAC2,INT2,FAC,LIM2,I1)                      SRTT
      IF(NOE.LT.3) RETURN                                               SRTT
      DO 9 JC=1,NOE                                                     SRTT
      IF((JC.EQ.JA).OR.(JC.EQ.JB)) GO TO 9                              SRTT
      CALL THREE(SLASHO,JA,JB,JC,FAC3,INT3,FAC,LIM3,I1)                 SRTT
      IF(NOE.LT.4) GO TO 9                                              SRTT
      JDL=JC+1                                                          SRTT
      IF(JDL.GT.NOE)GO TO 9                                             SRTT
      DO 10 JD=JDL,NOE                                                  SRTT
      IF((JD.EQ.JA).OR.(JD.EQ.JB).OR.(JD.EQ.JC)) GO TO 10               SRTT
      CALL FOUR(SLASHO,JA,JB,JC,JD,FAC4,INT4,FAC,LIM4,I1)               SRTT
   10 CONTINUE                                                          SRTT
    9 CONTINUE                                                          SRTT
      RETURN                                                            SRTT
    3 JA=DIFORB(1)                                                      SRTT
      JB=DIFORB(2)                                                      SRTT
      JC=DIFORB(3)                                                      SRTT
      CALL THREE(SLASHO,JA,JB,JC,FAC3,INT3,FAC,LIM3,I1)                 SRTT
      IF(NOE.LT.4) RETURN                                               SRTT
      DO 11 JD=1,NOE                                                    SRTT
      IF((JD.EQ.JA).OR.(JD.EQ.JB).OR.(JD.EQ.JC))GOTO 11                 SRTT
      CALL FOUR(SLASHO,JA,JB,JC,JD,FAC4,INT4,FAC,LIM4,I1)               SRTT
   11 CONTINUE                                                          SRTT
      RETURN                                                            SRTT
    4 CALL FOUR(SLASHO,DIFORB(1),DIFORB(2),DIFORB(3),DIFORB(4),FAC4,INT4SRTT
     1,FAC,LIM4,I1)                                                     SRTT
      RETURN                                                            SRTT
      END                                                               SRTT
```

```
      SUBROUTINE ONE(SLASHO,I,FAC1,INT1,FAC,LIM1,I1)                    ONET
      REAL*8 FAC1(3,50),FAC(3)                                          ONET
      INTEGER SLASHO(2,10),INT1(50)                                     ONET
      KB=IABS(SLASHO(1,I))                                              ONET
      DO 1 JA=1,LIM1                                                    ONET
      LI=JA                                                             ONET
      IF (INT1(JA).EQ.KB) GO TO 2                                       ONET
    1 CONTINUE                                                          ONET
      LIM1=LIM1+1                                                       ONET
      IF (50.LT.LIM1) GO TO 3                                           ONET
      INT1(LIM1)=KB                                                     ONET
      DO 4 JA=1,I1                                                      ONET
    4 FAC1(JA,LIM1)=FAC(JA)                                             ONET
      RETURN                                                            ONET
    2 DO 5 JA=1,I1                                                      ONET
    5 FAC1(JA,LI)=FAC1(JA,LI)+FAC(JA)                                   ONET
      RETURN                                                            ONET
    3 WRITE(8,900)                                                      ONET
      STOP                                                              ONET
  900 FORMAT('0',131('*')/'MORE THAN 50 ONE-ELE INTEGRALS'/131('*'))    ONET
      END                                                               ONET
      SUBROUTINE TWO(SLASHO,I,J,FAC2,INT2,FAC,LIM2,I1)                  TWOT
      REAL*8 FAC2(3,100),FAC(3),SIGN                                    TWOT
      INTEGER SLASHO(2,10),INT2(4,100),IV(2)                            TWOT
      INTEGER IX(4,4)/1,2,3,4,3,4,1,2,2,1,4,3,4,3,2,1/                  TWOT
      LOGICAL SPIN                                                      TWOT
      IJN1(I,J)=MINO(I,J)+MAXO(I,J)*(MAXO(I,J)-1)/2                     TWOT
      SPIN(I,J,K,L)=(0.GT.ISIGN(1,I)*ISIGN(1,J)).OR.(0.GT.ISIGN(1,K)*   TWOT
     1ISIGN(1,L))                                                       TWOT
      I1B=IABS(SLASHO(1,I))                                             TWOT
      I2B=IABS(SLASHO(1,J))                                             TWOT
      SIGN=1.D0                                                         TWOT
      DO 1 JA=1,2                                                       TWOT
      DO 2 JB=1,2                                                       TWOT
      IF(JB.EQ.JA) GO TO 2                                              TWOT
      IF(JB.EQ.1) SIGN=-SIGN                                            TWOT
      IV(JA)=I                                                          TWOT
      IV(JB)=J                                                          TWOT
      IF(SPIN(SLASHO(1,I),SLASHO(2,IV(1)),SLASHO(1,J),SLASHO(2,IV(2)))  TWOT
     1)GO TO 2                                                          TWOT
      I1K=IABS(SLASHO(2,IV(1)))                                         TWOT
      I2K=IABS(SLASHO(2,IV(2)))                                         TWOT
      DO 3 JC=1,LIM2                                                    TWOT
      DO 6 JD=1,4                                                       TWOT
      IF(I1B.NE.INT2(IX(1,JD),JC))GOTO6                                 TWOT
      IF(I1K.NE.INT2(IX(2,JD),JC))GOTO6                                 TWOT
      IF(I2B.NE.INT2(IX(3,JD),JC))GOTO6                                 TWOT
      IF(I2K.EQ.INT2(IX(4,JD),JC))GOTO7                                 TWOT
    6 CONTINUE                                                          TWOT
      GOTO3                                                             TWOT
      DO 4 JD=1,I1                                                      TWOT
    4 FAC2(JD,JC)=FAC2(JD,JC)+FAC(JD)*SIGN                              TWOT
      GO TO 2                                                           TWOT
    3 CONTINUE                                                          TWOT
      LIM2=LIM2+1                                                       TWOT
      IF(100.LT.LIM2)GOTO10                                             TWOT
      INT2(1,LIM2)=I1B                                                  TWOT
      INT2(2,LIM2)=I1K                                                  TWOT
      INT2(3,LIM2)=I2B                                                  TWOT
      INT2(4,LIM2)=I2K
```

```
      DO 5 JD=1,I1                                                      TWOT
    5 FAC2(JD,LIM2)=FAC(JD)*SIGN                                        TWOT
    2 CONTINUE                                                          TWOT
    1 CONTINUE                                                          TWOT
      RETURN                                                            TWOT
   10 WRITE(8,900)                                                      TWOT
  900 FORMAT('0',131('*')/20X,'MORE THAN 50 TWO-ELE INTEGRALS'/131('*'))TWOT
      STOP                                                              TWOT
      END                                                               TWOT
      SUBROUTINE THREE(SL,I,J,K,FAC3,INT3,FAC,L3,I1)                    THRT
      REAL*8 FAC3(3,200),FAC(3),SIGN,SIG                                THRT
      INTEGER SL(2,10),INT3(6,200),IV(3)                                THRT
      INTEGER IX(6,12)/1,2,3,4,5,6,3,4,1,2,5,6,5,6,3,4,1,2,1,2,5,6,3,4,5THRT
     .,6,1,2,3,4,3,4,5,6,1,2,2,1,4,3,6,5,4,3,2,1,6,5,6,5,4,3,2,1,2,1,6,5THRT
     .,4,3,6,5,2,1,4,3,4,3,6,5,2,1/                                     THRT
      LOGICAL SPIN                                                      THRT
      IJN1(I,J)=MIN0(I,J)+MAX0(I,J)*(MAX0(I,J)-1)/2                     THRT
      SPIN(I,J,K,L,M,N)= (0.GT.ISIGN(1,I)*ISIGN(1,J)).OR.(0.GT.ISIGN(1,KTHRT
     1)*ISIGN(1,L)).OR.(0.GT.ISIGN(1,M)*ISIGN(1,N))                     THRT
      I1B=IABS(SL(1,I))                                                 THRT
      I2B=IABS(SL(1,J))                                                 THRT
      I3B=IABS(SL(1,K))                                                 THRT
      DO 2 JA=1,3                                                       THRT
      DO 3 JB=1,3                                                       THRT
      IF(JA.EQ.JB) GO TO 3                                              THRT
      DO 4 JC=1,3                                                       THRT
      IF((JC.EQ.JA).OR.(JC.EQ.JB)) GO TO 4                              THRT
      IV(JA)=I                                                          THRT
      IV(JB)=J                                                          THRT
      IV(JC)=K                                                          THRT
      IF (SPIN(SL(1,I),SL(2,IV(1)),SL(1,J),SL(2,IV(2)),SL(1,K),SL(2,IV(3THRT
     1)))) GO TO 4                                                      THRT
      SIGN=SIG(3,JA,JB,JC,4)                                            THRT
      I1K=IABS(SL(2,IV(1)))                                             THRT
      I2K=IABS(SL(2,IV(2)))                                             THRT
      I3K=IABS(SL(2,IV(3)))                                             THRT
      DO 5 JD=1,L3                                                      THRT
      DO 8 JE=1,12                                                      THRT
      IF(I1B.NE.INT3(IX(1,JE),JD))GOTO8                                 THRT
      IF(I1K.NE.INT3(IX(2,JE),JD))GOTO8                                 THRT
      IF(I2B.NE.INT3(IX(3,JE),JD))GOTO8                                 THRT
      IF(I2K.NE.INT3(IX(4,JE),JD))GOTO8                                 THRT
      IF(I3B.NE.INT3(IX(5,JE),JD))GOTO8                                 THRT
      IF(I3K.EQ.INT3(IX(6,JE),JD))GOTO9                                 THRT
    8 CONTINUE                                                          THRT
      GOTO5                                                             THRT
    9 DO 6 JE=1,I1                                                      THRT
    6 FAC3(JE,JD)=FAC3(JE,JD)+FAC(JE)*SIGN                              THRT
      GO TO 4                                                           THRT
    5 CONTINUE                                                          THRT
      L3=L3+1                                                           THRT
      IF(200.LT.L3)GOTO10                                               THRT
      INT3(1,L3)=I1B                                                    THRT
      INT3(2,L3)=I1K                                                    THRT
      INT3(3,L3)=I2B                                                    THRT
      INT3(4,L3)=I2K                                                    THRT
      INT3(5,L3)=I3B                                                    THRT
      INT3(6,L3)=I3K                                                    THRT
      DO 7 JE=1,I1                                                      THRT
    7 FAC3(JE,L3)=FAC(JE)*SIGN
```

```
  4 CONTINUE                                                            THRT
  3 CONTINUE                                                            THRT
  2 CONTINUE                                                            THRT
    RETURN                                                              THRT
 10 WRITE(8,900)                                                        THRT
    STOP                                                                THRT
900 FORMAT('0',131('*')/30X,'MORE THAN 50 THREE*ELE INTEGRALS'/131('*'THRT
   1))                                                                  THRT
    END                                                                 THRT
    SUBROUTINE FOUR(SL,I,J,K,L,FAC4,INT4,FAC,L4,I1)                     FORT
    REAL*8 FAC4(3,300),FAC(3),SIGN,SIG                                  FORT
    INTEGER SL(2,10),INT4(8,300),IV(4)                                  FORT
    INTEGER IX(8,48)/1,2,3,4,5,6,7,8,3,4,1,2,5,6,7,8,5,6,3,4,1,2,7,8,7FORT
   .,8,3,4,5,6,1,2,1,2,5,6,3,4,7,8,1,2,7,8,5,6,3,4,1,2,3,4,7,8,5,6,5,6FORT
   .,1,2,3,4,7,8,3,4,5,6,1,2,7,8,7,8,1,2,5,6,3,4,3,4,7,8,5,6,1,2,7,8,3FORT
   .,4,1,2,5,6,5,6,3,4,7,8,1,2,1,2,7,8,3,4,5,6,1,2,5,6,7,8,3,4,7,8,1,2FORT
   .,3,4,5,6,7,8,5,6,1,2,3,4,5,6,1,2,7,8,3,4,3,4,7,8,1,2,5,6,5,6,7,8,3FORT
   .,4,1,2,3,4,5,6,7,8,1,2,3,4,1,2,7,8,5,6,5,6,7,8,1,2,3,4,7,8,5,6,3,4FORT
   .,1,2,2,1,4,3,6,5,8,7,4,3,2,1,6,5,8,7,6,5,4,3,2,1,8,7,8,7,4,3,6,5,2FORT
   .,1,2,1,6,5,4,3,8,7,2,1,8,7,6,5,4,3,2,1,4,3,8,7,6,5,6,5,2,1,4,3,8,7FORT
   .,4,3,6,5,2,1,8,7,8,7,2,1,6,5,4,3,4,3,8,7,6,5,2,1,8,7,4,3,2,1,6,5,6FORT
   .,5,4,3,8,7,2,1,2,1,8,7,4,3,6,5,2,1,6,5,8,7,4,3,8,7,2,1,4,3,6,5,8,7FORT
   .,6,5,2,1,4,3,6,5,2,1,8,7,4,3,4,3,8,7,2,1,6,5,6,5,8,7,4,3,2,1,4,3,6FORT
   .,5,8,7,2,1,4,3,2,1,8,7,6,5,6,5,8,7,2,1,4,3,8,7,6,5,4,3,2,1/        FORT
    LOGICAL SPIN                                                        FORT
    IJN1(I,J)=MIN0(I,J)+MAX0(I,J)*(MAX0(I,J)-1)/2                       FORT
    SPIN(IB,IK,JZ,JK,KB,KK,LB,LK)= (0.GT.ISIGN(1,IB)*ISIGN(1,IK)).OR.  FORT
   1(0.GT.ISIGN(1,JZ)*ISIGN(1,JK)).OR.(0.GT.ISIGN(1,KB)*ISIGN(1,KK)).OFORT
   2R.(0.GT.ISIGN(1,LB)*ISIGN(1,LK))                                    FORT
    I1B=IABS(SL(1,I))                                                   FORT
    I2B=IABS(SL(1,J))                                                   FORT
    I3B=IABS(SL(1,K))                                                   FORT
    I4B=IABS(SL(1,L))                                                   FORT
    DO 1 JA=1,4                                                         FORT
    DO 2 JB=1,4                                                         FORT
    IF(JA.EQ.JB) GO TO 2                                                FORT
    DO 3 JC=1,4                                                         FORT
    IF((JC.EQ.JA).OR.(JC.EQ.JB)) GO TO 3                                FORT
    DO 4 JD=1,4                                                         FORT
    IF((JD.EQ.JA).OR.(JD.EQ.JB).OR.(JD.EQ.JC)) GO TO 4                  FORT
    IV(JA)=I                                                            FORT
    IV(JB)=J                                                            FORT
    IV(JC)=K                                                            FORT
    IV(JD)=L                                                            FORT
    IF(SPIN(SL(1,I),SL(2,IV(1)),SL(1,J),SL(2,IV(2)),SL(1,K),SL(2,IV(3)FORT
   1),SL(1,L),SL(2,IV(4)))) GO TO 4                                     FORT
    SIGN=SIG(4,JA,JB,JC,JD)                                             FORT
    I1K=IABS(SL(2,IV(1)))                                               FORT
    I2K=IABS(SL(2,IV(2)))                                               FORT
    I3K=IABS(SL(2,IV(3)))                                               FORT
    I4K=IABS(SL(2,IV(4)))                                               FORT
    DO 5 JE=1,L4                                                        FORT
    DO 8 JF=1,48                                                        FORT
    IF(I1B.NE.INT4(IX(1,JF),JE))GOTO8                                   FORT
    IF(I1K.NE.INT4(IX(2,JF),JE))GOTO8                                   FORT
    IF(I2B.NE.INT4(IX(3,JF),JE))GOTO8                                   FORT
    IF(I2K.NE.INT4(IX(4,JF),JE))GOTO8                                   FORT
    IF(I3B.NE.INT4(IX(5,JF),JE))GOTO8                                   FORT
    IF(I3K.NE.INT4(IX(6,JF),JE))GOTO8                                   FORT
    IF(I4B.NE.INT4(IX(7,JF),JE))GOTO8                                   FORT
```

```
      IF(I4K.EQ.INT4(IX(8,JF),JE))GOTO9                                  FORT
8     CONTINUE                                                           FORT
      GOTO5                                                              FORT
9     DO 6 JF=1,I1                                                       FORT
    6 FAC4(JF,JE)=FAC4(JF,JE)+FAC(JF)*SIGN                               FORT
      GO TO 4                                                            FORT
    5 CONTINUE                                                           FORT
      L4=L4+1                                                            FORT
      IF(300.LT.L4)GOTO10                                                FORT
      INT4(1,L4)=I1B                                                     FORT
      INT4(2,L4)=I1K                                                     FORT
      INT4(3,L4)=I2B                                                     FORT
      INT4(4,L4)=I2K                                                     FORT
      INT4(5,L4)=I3B                                                     FORT
      INT4(6,L4)=I3K                                                     FORT
      INT4(7,L4)=I4B                                                     FORT
      INT4(8,L4)=I4K                                                     FORT
      DO 7 JF=1,I1                                                       FORT
    7 FAC4(JF,L4)=FAC(JF)*SIGN                                           FORT
    4 CONTINUE                                                           FORT
    3 CONTINUE                                                           FORT
    2 CONTINUE                                                           FORT
    1 CONTINUE                                                           FORT
      RETURN                                                             FORT
   10 WRITE(8,900)                                                       FORT
  900 FORMAT('0',131('*')/20X,'MORE THAN 50 FOUR-ELF-INTEGRALS'/131('*')FORT
     1)                                                                  FORT
      STOP                                                               FORT
      END                                                                FORT
      FUNCTION SIG(N,I,J,K,L)                                            FSIG
      REAL*8 SIG                                                         FSIG
      INTEGER IV(4)                                                      FSIG
      IV(1)=I                                                            FSIG
      IV(2)=J                                                            FSIG
      IV(3)=K                                                            FSIG
      IV(4)=L                                                            FSIG
      ISUM=0                                                             FSIG
      NM1=N-1                                                            FSIG
      DO 1 JA=1,NM1                                                      FSIG
      JA1=JA+1                                                           FSIG
      DO 1 JB=JA1,N                                                      FSIG
    1 IF(IV(JA).GT.IV(JB)) ISUM=ISUM+1                                   FSIG
      SIG=1.D0*DFLOAT((-1)**ISUM)                                        FSIG
      RETURN                                                             FSIG
      END                                                                FSIG
      SUBROUTINE VECT(IVEC,ISTA,ICOMV,M,CONFIG,N)                        VECT
      INTEGER*2 IVEC(10),ISTA(10),ICOMV(10),CONFIG(33),ICO1A(10),        VECT
     1ICO1B(10)/10*1/,ICO1(10)                                           VECT
      DO 11 J1 = 1,10                                                    VECT
11    ICO1B(J1) = 1                                                      VECT
      M3 = M/3                                                           VECT
      DO 10 J1 = 1,M3                                                    VECT
10    ICO1(J1)=(2*CONFIG(J1*3-1)+1)*2                                    VECT
      ICO1A(1) = ICO1(1)                                                 VECT
      DO 20 J1 = 2,M3                                                    VECT
      ICO1A(J1) = ICO1A(J1-1) + ICO1(J1)                                 VECT
20    ICO1B(J1) = ICO1B(J1-1) + ICO1(J1-1)                               VECT
      J3 = 1                                                             VECT
      DO 30 J1 = 1,M3                                                    VECT
      J11 = CONFIG(J1*3)
```

```
      DO 31 J2 = 1,J11                                                  VECT
      IVEC(J3) = IC01B(J1) + J2 - 1                                     VECT
      ISTA (J3) = IVEC(J3)                                              VECT
 31   J3 = J3 + 1                                                       VECT
 30   CONTINUE                                                          VECT
      J3 = N                                                            VECT
      DO 40 J1 = 1,M3                                                   VECT
      J11 = CONFIG((M3-J1+1)*3)                                         VECT
      DO 41 J2 = 1,J11                                                  VECT
      ICOMV(J3) = IC01A(M3-J1+1)-J2+2                                   VECT
 41   J3 = J3-1                                                         VECT
 40   CONTINUE                                                          VECT
      RETURN                                                            VECT
      END                                                               VECT
      SUBROUTINE EXPAND (STATE,CONFIG,DMAT,N,M)                         EXPD
C THIS ROUTINE EXPANDS THE CONFIGURATION INTO ALL POSSIBLE STATES       EXPD
C FROM WHICH THE SINGLE DETERMINANTS ARE CHOSEN                         EXPD
C CVEC CONTAINS THE POSITION OF THE BEGINNING OF A NEW SHELL IN DMAT    EXPD
      INTEGER*2 STATE(2),CONFIG( 33),DMAT(4,100),CVEC(10)               EXPD
      J1=1                                                              EXPD
      J2=0                                                              EXPD
      J3=0                                                              EXPD
      J4 = M/3                                                          EXPD
      DO 10  I=1,J4                                                     EXPD
      J1=J1+J2                                                          EXPD
      J2 = 2*(2* CONFIG(3*I-1)+1)                                       EXPD
      J3 = J3 + J2                                                      EXPD
      DO 10 J= J1,J3                                                    EXPD
      DMAT (1,J) = CONFIG(3*I-2)                                        EXPD
      DMAT (2,J) = CONFIG(3*I-1)                                        EXPD
      DMAT (3,J) = CONFIG (3*I-1) - (J-J1)/2                            EXPD
   10 DMAT (4,J) = (-1)**J                                              EXPD
      RETURN                                                            EXPD
      END                                                               EXPD
      SUBROUTINE RESET  (IVEC,I,*,*,M,.ISTA,*)                          RSET
      INTEGER*2 IVEC(20),ISTA(20)                                       RSET
      IF (I.EQ.1) RETURN 2                                              RSET
      IVEC(I-1) = IVEC (I-1) + 1                                        RSET
      DO 10 J=I,M                                                       RSET
   10 IVEC (J) = ISTA (J)                                               RSET
      DO 20 J = I,  M                                                   RSET
   20 IF ( IVEC (J-1) .GE. IVEC(J)) IVEC (J) = IVEC (J-1)+1             RSET
      IF (I.LT.M) RETURN3                                               RSET
      RETURN 1                                                          RSET
      END                                                               RSET
      SUBROUTINE DETVAR (DMAT,IVEC,SLDV,N,STATE,K,NUMDET,*)             DETV
      INTEGER*2 DMAT (4,100),IVEC(20),SLDV( 52,4,20),STATE(2),NUMDET(52,DETV
     120)                                                               DETV
      SUM1=0                                                            DETV
      SUM2=1                                                            DETV
      DO 10 J=1,N                                                       DETV
      SUM1 = SUM1 + DMAT(3,IVEC(J))                                     DETV
   10 SUM2 = SUM2 + DMAT(4,IVEC(J))                                     DETV
      IF ( (SUM1.NE.STATE(2)).OR.(SUM2.NE.STATE(1))) RETURN             DETV
      K=K+1                                                             DETV
      IF (K .LE. 52) GO TO 11                                           DETV
      WRITE(6,900) K                                                    DETV
      RETURN 1                                                          DETV
  900 FORMAT('1   THERE ARE MORE THAN',I4,'  SLATERDETERMINANTS')       DETV
   11 DO 20  J2 = 1,N
```

```
      NUMDET (K,J2) = IVEC (J2)                                          DETV
      DO 20  J1 = 1,4                                                    DETV
   20 SLDV (K,J1,J2) = DMAT(J1,IVEC(J2))                                 DETV
      RETURN                                                             DETV
      END                                                                DETV
      SUBROUTINE CHECK (IVEC,ISTA,*,N,*,ICOMV)                           CHCK
      INTEGER*2 IVEC (20),ISTA (20),ICOMV(20)                            CHCK
   39 CONTINUE                                                           CHCK
      DO 10 J=1,N                                                        CHCK
      IF (  IVEC(J) .GE. ICOMV(J)) CALL RESET(IVEC,J,&10,&20,N,ISTA,&39)CHCK
   10 CONTINUE                                                           CHCK
      RETURN1                                                            CHCK
   20 RETURN 2                                                           CHCK
      END                                                                CHCK
      SUBROUTINE OPERAT (SLDV ,LSSQMA,K,STATE,CMAT,N,EIGVEC,B,*,I1)      OPER
C THIS ROUTINE OPERATES WITH L-SQUARE AND S-SQUARE ON THE SLATORS        OPER
C (WHICH ARE CONTAINED IN SLDV).  IT SETS UP THE MATRIX LSSQMA           OPER
C WHICH WILL BE DIAGONALIZED TO GIVE THE REQUIRED EIGENVALUES AND        OPER
C EIGENVECTORS                                                           OPER
C                                                                        OPER
      INTEGER*2 SLDV ( 52,4,20),CMAT(4,20), STATE(2)                     OPER
      REAL*8    LSSQMA ( K,K ),EIGVEC(K,K),B(K),S2                       OPER
      IALPHA = N/2 + STATE(1)/2                                          OPER
      IBETA = N - IALPHA                                                 OPER
      S2 = (1.25D-2)*DFLOAT(2*(IALPHA+IBETA)+(IALPHA-IBETA)**2)          OPER
      DO 20 J=1,K                                                        OPER
      DO 10 I1=1,N                                                       OPER
      DO 10 I2=1,4                                                       OPER
   10 CMAT (I2,I1) = SLDV (J,I2,I1)                                      OPER
   20 CALL LSSQUA ( CMAT, SLDV, K, LSSQMA, STATE, J,N,S2)                OPER
C                                                                        OPER
C  THE LOOP 60 COMPRESSES LSSQMA, SO THAT IT CAN BE HANDLED              OPER
C  BY THE SUBROUTINE DEIGE                                               OPER
C                                                                        OPER
      K1 = K-1                                                           OPER
      J3 = 1                                                             OPER
      J4 = 1                                                             OPER
      DO 60 J1 = 1,K1                                                    OPER
      DO 60 J2 = 1,K                                                     OPER
      LSSQMA(J2,J1) = LSSQMA(J3,J4)                                      OPER
      J3 = J3 + 1                                                        OPER
      IF (J3 .LE. J4) GO TO 60                                           OPER
      J4 = J4 + 1                                                        OPER
      IF (J4 .GT. K) GO TO 61                                            OPER
      J3 = 1                                                             OPER
   60 CONTINUE                                                           OPER
      IF (K .EQ. 2) LSSQMA(1,2) = LSSQMA (2,2)                           OPER
C                                                                        OPER
C  DEIGE IS A REAL*8 JACOBI DIAGONALIZATION ADAPTED FROM SSP             OPER
C                                                                        OPER
   61 CALL DEIGE(LSSQMA,EIGVEC,K,0)                                      OPER
C                                                                        OPER
C                                                                        OPER
C  THE LOOP 70 PICKS OUT THE E'VALUES OF LSSQMA                          OPER
C                                                                        OPER
      J3 = 1                                                             OPER
      J4 = 1                                                             OPER
      DO 70 J1 = 1,K                                                     OPER
      DO 70 J2 = 1,K                                                     OPER
      IF (J3 .EQ. J4) B(J4) = LSSOMA(J2,J1)
```

```
      J3 = J3 + 1                                                       OPER
      IF (J3 .LE. J4) GO TO 70                                          OPER
      J4 = J4 + 1                                                       OPER
      J3 = 1                                                            OPER
   70 CONTINUE                                                          OPER
C                                                                       OPER
C   SHREIB PRINTS OUT THE APPROPRIATE E'VECTORS                         OPER
C                                                                       OPER
      CALL SHREIB(EIGVEC,B,K,STATE,&100,I1)                             OPER
C                                                                       OPER
      RETURN                                                            OPER
  100 RETURN1                                                           OPER
      END                                                               OPER
      SUBROUTINE LSSQUA (CMAT, SLDV, K,LSSQMA, STATE,J1,N,S2)           LSSQ
C                                                                       LSSQ
C  THIS SUBROUTINE DETERMINES THE ELEMENTS OF LSSQMA                    LSSQ
C                                                                       LSSQ
      REAL*8   LSSQMA ( K,K ),F1,F2    ,S2 ,DFLOAT                      LSSQ
      INTEGER*2 CMAT (4,20),SLDV ( 52,4,20),STATE(2)                    LSSQ
      LSSQMA(J1,J1) = S2                                                LSSQ
C                                                                       LSSQ
C  NOW THE EXCHANGE OPERATOR IS APPLIED ON CMAT                         LSSQ
C                                                                       LSSQ
      DO 20 I20 = 1,N                                                   LSSQ
      DO 20 I21 = I20,N                                                 LSSQ
      IF ( CMAT (4,I20) .EQ. CMAT (4,I21)) GO TO 20                     LSSQ
      CMAT (4,I20) = CMAT(4,I20)*(-1)                                   LSSQ
      CMAT (4,I21) = CMAT (4,I21)* (-1)                                 LSSQ
      CALL COMP (CMAT,K,SLDV,LSSQMA,4,J1,F1,F2,N)                       LSSQ
      CMAT (4,I20) = CMAT(4,I20)*(-1)                                   LSSQ
      CMAT (4,I21) = CMAT (4,I21)*(-1)                                  LSSQ
   20 CONTINUE                                                          LSSQ
C                                                                       LSSQ
C  NOW THE L-SQUARE-PART IS COMPUTED                                    LSSQ
C                                                                       LSSQ
      IS = STATE(2)*(STATE(2)+1)                                        LSSQ
      LSSQMA(J1,J1)=LSSQMA(J1,J1)+DFLOAT(IS)                            LSSQ
      DO 30 I30 = 1,N                                                   LSSQ
      CMAT (3,I30) = CMAT (3,I30) +1                                    LSSQ
      IF  (CMAT (3,I30) .GT. CMAT (2,I30)) GO TO 33                     LSSQ
      IJ3 = CMAT (3,I30)-1                                              LSSQ
      DO 31 I31 = 1,N                                                   LSSQ
      CMAT(3,I31) = CMAT(3,I31) -1                                      LSSQ
      IF (CMAT (2,I31) .LT. (-1)*CMAT(3,I31)) GO TO 32                  LSSQ
      IJ1 = CMAT (2,I30)                                                LSSQ
      IJ2 = CMAT (2,I31)                                                LSSQ
      IJ4 = CMAT (3,I31) + 1                                            LSSQ
      F1 = DFLOAT ( IJ1*( IJ1+1) - IJ3*(IJ3+1))                         LSSQ
      F2 = DFLOAT ( IJ2*(IJ2+1) - IJ4 *(IJ4-1))                         LSSQ
      CALL COMP (CMAT,K,SLDV,LSSQMA,3,J1,F1,F2,N)                       LSSQ
   32 CMAT(3,I31) = CMAT(3,I31)+1                                       LSSQ
   31 CONTINUE                                                          LSSQ
   33 CMAT(3,I30) = CMAT(3,I30)-1                                       LSSQ
   30 CONTINUE                                                          LSSQ
      RETURN                                                            LSSQ
      END                                                               LSSQ
      SUBROUTINE  COMP ( MA,K,SLDV,LS,I,KL,F1,F2,N)                     COMP
C                                                                       COMP
C  THIS ROUTINE COMPARES THE SLATOR MA WITH THE SLATORS IN SLDV AND     COMP
C  ASSIGNS APPROPRIATE MATRIX-ELEMENTS OF LS                            COMP
```

```
C
      INTEGER AUSFAL(10)                                                COMP
      INTEGER*2 MA (4,20),SLDV ( 52,4,20)                               COMP
      REAL*8 LS ( K,K ),F1,F2 ,DSQRT,  DFLOAT                           COMP
      DO 10 I1 = 1,K                                                    COMP
      DO 11 I21 = 1,10                                                  COMP
   11 AUSFAL(I21) = 0                                                   COMP
      INDIC=1                                                           COMP
      ISIGN = 0                                                         COMP
      DO 30 I3 = 1,N                                                    COMP
      DO 20 I2 = 1,N                                                    COMP
      DO 21 I21 =1,INDIC                                                COMP
      IF(I2 .EQ. AUSFAL(I21)) GO TO 20                                  COMP
   21 CONTINUE                                                          COMP
      DO 40 I4 = 1,4                                                    COMP
      IF ( MA  (I4,I3) .NE. SLDV (I1,I4,I2)) GO TO 20                   COMP
   40 CONTINUE                                                          COMP
      AUSFAL(INDIC)=I2                                                  COMP
      INDIC=INDIC+1                                                     COMP
      IF(I3.NE.I2)ISIGN=ISIGN+1                                         COMP
      GO TO 30                                                          COMP
   20 CONTINUE                                                          COMP
      GO TO 10                                                          COMP
   30 CONTINUE                                                          COMP
      GO TO 1                                                           COMP
   10 CONTINUE                                                          COMP
      RETURN                                                            COMP
    1 IF(ISIGN.NE.0)ISIGN=ISIGN+1                                       COMP
      IF(I.EQ.3)GO TO 2                                                 COMP
      LS (I1,KL) = LS (I1,KL)+(5.D-2)*DFLOAT ((-1)**MOD(ISIGN,2))       COMP
      RETURN                                                            COMP
    2 LS (I1,KL) = LS(I1,KL) + DSQRT(F1*F2)*DFLOAT((-1)**MOD(ISIGN,2))  COMP
      RETURN                                                            COMP
      END                                                               COMP
      SUBROUTINE SHREIB(MAT,EVAL,K,STATE,*,I1)                          SHRB
C                                                                       SHRB
C     THIS ROUTINE WRITES OUT THE E'VECTORS FOR THE                     SHRB
C     GIVEN TERM AND CONFIGURATION                                      SHRB
C                                                                       SHRB
      REAL*8 MAT(K,K),EVAL(K),ST1,ST2                                   SHRB
      INTEGER*2 STATE(2)                                                SHRB
      IST1 = STATE(2)*(STATE(2)+1)                                      SHRB
      IST2 = STATE(1) -1                                                SHRB
      ST1 = DFLOAT(IST1) + 1.25D-2*DFLOAT(IST2* (IST2+2))               SHRB
      I1 = 0                                                            SHRB
      DO 10 J1 = 1,K                                                    SHRB
      IF (DABS(ST1-EVAL(K-J1+1)) .GT. 1.D-9) GO TO 11                   SHRB
      I1 = I1 + 1                                                       SHRB
   10 CONTINUE                                                          SHRB
   11 IF (I1 .EQ. 0) GO TO 12                                           SHRB
      WRITE(6,900) I1                                                   SHRB
      DO 20 J1 = 1,I1                                                   SHRB
      ST2 = DABS(ST1-EVAL(K-J1+1))                                      SHRB
   20 WRITE(6,901) J1,ST2,(J2,MAT(J2,(K-J1+1)),J2=1,K)                  SHRB
      RETURN                                                            SHRB
   12 WRITE (6,902)                                                     SHRB
      RETURN1                                                           SHRB
  900 FORMAT (//////'     THERE EXIST',I4,'  LINEARLY INDEPENDENT EIGENFSHRB
     1UNCTION(S)')                                                      SHRB
  901 FORMAT (///' EIGENVECTOR',I4,'.',60X,'EIGENVALUE ROUND-OFF ERROR  SHRB
```

```
   1'    ,1PD10.1 //10(4(4X,I3,')',0PD25.15)/))
 902 FORMAT(////' THERE EXISTS NO STATE WITH THE GIVEN ML AND MS VALUESSHRB
    1'/' FOR THE ABOVE CONFIGURATION')                                  SHRB
     END                                                                SHRB
     SUBROUTINE OUTPU(MAT,STATE,CONF,K,N,*,ID,STTE)                     SHRB
C                                                                       OUT1
C   THIS ROUTINE PRINTS                                                 OUT1
C      THE NUMBER OF ELECTRONS                                          OUT1
C      THE CONFIGURATION                                                OUT1
C      THE VALUES OF ML AND MS.                                         OUT1
C                                                                       OUT1
     INTEGER*2 MAT(52,4,10),STATE(2),CONF(33),COW(8)/'S ','P ','D ','F OUT1
    1','G ','H ','I ','K '/,  LINE(22)/22*'  '/,IT1,IT2,STTE(2)         OUT1
     REAL*8 SPIN(11),ALPH/'ALPHA   '/,BET/' BETA    '/                  OUT1
     N2 = N*2                                                           OUT1
     IK=1                                                               OUT1
C                                                                       OUT1
C   THE NEXT STMTS FILL THE VARIABLE 'LINE' UP WITH                     OUT1
C   THE PRINTOUT FOR THE CONFIGURATION                                  OUT1
C                                                                       OUT1
     JK = 2                                                             OUT1
  31 INCR=0                                                             OUT1
  30 LINE(IK) = CONF(JK-1)                                              OUT1
     LINE(IK+1) = COW(CONF(JK)+1)                                       OUT1
     IK = IK +2                                                         OUT1
     INCR = INCR +1                                                     OUT1
C                                                                       OUT1
C  THIS STMT CHECKS IF ALL THE ORBITALS FOR EACH ELECTRON               OUT1
C   HAVE BEEN EXHAUSTED.                                                OUT1
C                                                                       OUT1
     IF (INCR .NE. CONF(JK+1)) GO TO 30                                 OUT1
     JK = JK + 3                                                        OUT1
     IF (IK .LT.N2) GO TO 31                                            OUT1
C                                                                       OUT1
C   IT1 AND TI2 CONTAIN THE VALUE OF ML AND MS.                         OUT1
C   MS CAN BE HALF INTEGRAL                                             OUT1
C                                                                       OUT1
     IT1 = STATE(2)                                                     OUT1
     IST1 =STATE(1)                                                     OUT1
     TI2 = FLOAT (IST1-1)/2.0                                           OUT1
     IT11 = STTE(2)                                                     OUT1
     IST2 = STTE(1)                                                     OUT1
     TI21 = FLOAT(IST2-IST1) + FLOAT(IST1-1)/2.0                        OUT1
     IF (ID .EQ. 1) GO TO 42                                            OUT1
     WRITE (6,910) N,(LINE(I),I=1,N2)                                   OUT1
     IF (K .NE. 0) GO TO 41                                             OUT1
     WRITE(6,914) IT1,TI2                                               OUT1
     RETURN1                                                            OUT1
  42 WRITE(6,915)                                                       OUT1
  41 WRITE(6,913) IT1,IT11,TI2,TI21,(LINE(I),I=1,N2)                    OUT1
     DO 33 IK = 1,K                                                     OUT1
     DO 34  IJ = 1,N                                                    OUT1
     IF (MAT(IK,4,IJ))35,35,36                                          OUT1
  35 SPIN (IJ) = BET                                                    OUT1
     GO TO 34                                                           OUT1
  36 SPIN(IJ) = ALPH                                                    OUT1
  34 CONTINUE                                                           OUT1
     WRITE (6,911) IK,(MAT(IK,3,IL),IL = 1,N)                           OUT1
  33 WRITE (6,912) (SPIN(IL),IL = 1,N)                                  OUT1
 910 FORMAT('1',5X,'L-S EIGENFUNCTIONS BY DIRECT DIAGONALIZATION'///6X,OUT1
```

```
    1 'THE NUMBER OF ELECTRONS IS',I5//6X,'THE ORBITAL OCCUPANCY IS',3XOUT1
    2,11(I2,A2))                                                      OUT1
911 FORMAT ('0',3X,I3,'.',3X,11(I2,'=ML',4X))                         OUT1
912 FORMAT (' ',10X,11(A8,1X))                                        OUT1
913 FORMAT(' '//5X,' L=',I2,'   ML=',I2,'      S=',F4.1,'   MS=',F4.1//OUT1
    1/5X,' THE POSSIBLE SLATERDETERMINANTS CORRESPONDING TO THE GIVEN OOUT1
    2RBITAL OCCUPANCY AND VALUES OF ML AND MS ARE'//' ',6X,11(5X,I2,A2 OUT1
    3))                                                               OUT1
914 FORMAT('-      L=ML=',I3,10X,'S=MS=',F5.1///' THERE EXISTS NO SLATOOUT1
    1R WITH THE GIVEN ML AND MS VALUES'/' FOR THE ABOVE CONFIGURATION')OUT1
915  FORMAT('1')                                                      OUT1
     RETURN                                                           OUT1
     END                                                              OUT1
     IMPLICIT REAL*8 (A-H,O-Z)                                        MAIN
     COMMON/ALL/EXPCOE(5,10),ORBEXP(15),H(5,5,3),CHARGE,QN,NOBT(3),ISYMMAIN
    .,FDUB                                                            MAIN
     INTEGER  INTNO2,QN(15),FDUB,INFO(4),ORB(3)                       MAIN
     COMMON/HINZ/S,F,L,NOB,NORB,CLOSED                                MAIN
     COMMON/ONE/FAC1(50),FHH1(5,5,4),FH1(5,5,4),INT1(50),LIM1         MAIN
     COMMON/TWO/FH2(5,5,4),FHH2(5,5,4),FAC2(100),LH2(5,5,4,4),LHH2(5,5,MAIN
    .4,4),INT2(4,100),INTNO2(100),NULL2(100),LIM2,INTLI2              MAIN
     COMMON/THREE/FAC3(200),FHH3(5,5,4),LHH3(5,5,4,4),INT3(6,200),INTNOMAIN
    .3(3,100),LIM3,INTLI3,NULL3                                       MAIN
     LOGICAL NULL2,NULL3(100),NULL4(100),CLOSED(3,4),LOGCOM(3)        MAIN
     COMMON/FOUR/FAC4(300),FHH4(5,5,4),LHH4(5,5,4,4),INT4(8,300),LIM4,NMAIN
    .ULL4                                                             MAIN
     REAL*8 F(5,5,4),LH2,LHH2,LHH3,LHH4,S(5,5,3),HH(5,5,3),L(5,5,4,4),OMAIN
    .ENER(3,4),EXHH(3,4),ENERGY(3,4),COMV(3),COMPLV(3)                MAIN
     CALL LOGIOU(INFO,'2        ',&100)                               MAIN
     FDUB=INFO(1)                                                     MAIN
12   CALL INPUT(ORB,NOBT,LIM1,LIM2,LIM3,LIM4,METHOD,INTLI2,INTLI3,QN,CHMAIN
    .ARGE,WK,CLOSED,ORBEXP,EXPCOE,INT1,FAC1,INT2,FAC2,INT3,FAC3,INT4,FAMAIN
    .C4,INTNO2,INTNO3,NULL2,NULL3,ITEFAC,IOPT,THRH,TAU)               MAIN
     ICOMPL=0                                                         MAIN
11   CALL ONEINT(HH,S)                                                MAIN
     IF(IOPT.LT.1)INTLI2=0                                            MAIN
     CALL OUT0(H,HH,S,EXPCOE,NOBT,ORB)                                MAIN
5    DO 20 ITER=1,ITEFAC                                              MAIN
     IF(ITER.NE.1)GOTO2                                               MAIN
     DO 1 JA=1,3                                                      MAIN
     LOGCOM(JA)=ORB(JA).EQ.0                                          MAIN
1    COMPLV(JA)=0.D0                                                  MAIN
     DO 10 JA=1,3                                                     MAIN
     IF(ORB(JA).EQ.0)GOTO10                                           MAIN
     CALL RENORM(NOBT(JA),ORB(JA),JA,EXPCOE,S)                        MAIN
10   CONTINUE                                                         MAIN
     GOTO4                                                            MAIN
2    COMV(1)=COMPLV(1)                                                MAIN
     COMV(2)=COMPLV(2)-COMV(1)                                        MAIN
     IF(COMV(2).LT.0.D0)COMV(2)=0.D0                                  MAIN
     COMV(3)=COMPLV(3)-COMV(2)-COMV(1)                                MAIN
     IF(COMV(3).LT.0.D0)COMV(3)=0.D0                                  MAIN
     DO 3 JA=1,3                                                      MAIN
3    LOGCOM(JA)=LOGCOM(JA).OR.(COMV(JA).LT.1.D-8)                     MAIN
4    LIMDI3=0                                                         MAIN
     LIMDI4=0                                                         MAIN
     COMPL=1.D0                                                       MAIN
     WRITE(8,900)ITER                                                 MAIN
     LIM21=1                                                          MAIN
     IF(ICOMPL.EQ.1)LIM21=2
```

```
      DO 21 ISYM=1,LIM21                                                  MAIN
      NORB=ORB(ISYM)                                                      MAIN
      IF(LOGCOM(ISYM))GOTO21                                              MAIN
      IF(NORB.EQ.0)GOTO21                                                 MAIN
      NOB=NOBT(ISYM)                                                      MAIN
      DO 30 JA=1,NOB                                                      MAIN
      DO 30 JB=1,NOB                                                      MAIN
      DO 30 JC=1,4                                                        MAIN
      FH1(JA,JB,JC)=0.D0                                                  MAIN
      FHH1(JA,JB,JC)=0.D0                                                 MAIN
      FH2(JA,JB,JC)=0.D0                                                  MAIN
      FHH2(JA,JB,JC)=0.D0                                                 MAIN
      FHH3(JA,JB,JC)=0.D0                                                 MAIN
      FHH4(JA,JB,JC)=0.D0                                                 MAIN
      DO 30 JD=1,4                                                        MAIN
      LH2(JA,JB,JC,JD)=0.D0                                               MAIN
      LHH2(JA,JB,JC,JD)=0.D0                                              MAIN
      LHH3(JA,JB,JC,JD)=0.D0                                              MAIN
30    LHH4(JA,JB,JC,JD)=0.D0                                              MAIN
      CALL OUT01(EXPCOE,NOB,NORB,ISYM)                                    MAIN
      CALL ONEEL(NOB,ISYM,H,HH)                                           MAIN
      CALL OUT1(FH1,FHH1,NORB,NOB,ISYM)                                   MAIN
      CALL TIME(1,1)                                                      MAIN
      CALL TWOELE                                                         MAIN
      CALL TIME(1,1)                                                      MAIN
      CALL OUT2(FH2,FHH2,LH2,LHH2,NORB,NOB,1,1,ISYM)                      MAIN
      IF(METHOD.EQ.1.OR.LIM3.EQ.0)GOTO31                                  MAIN
      CALL THREEL(LIMDI3,NORB,NOB)                                        MAIN
      CALL TIME(1,1)                                                      MAIN
      CALL OUT3(FHH3,LHH3,NORB,NOB,ISYM)                                  MAIN
      IF(LIM4.EQ.0)GOTO31                                                 MAIN
      CALL FOUREL(NORB,NOB,LIMDI4)                                        MAIN
      CALL TIME(1,1)                                                      MAIN
      CALL OUT4(FHH4,LHH4,NORB,NOB,1,ISYM)                                MAIN
31    CALL COMBIN(METHOD,ISYM,ORB,NOBT,FH1,FHH1,FH2,FHH2,FHH3,FHH4,LH2,LMAIN
     .HH2,LHH3,LHH4,WK,EXPCOE,ENERGY,EXHH,TAU)                            MAIN
      CALL TIME(1,1)                                                      MAIN
      CALL OUT4(F,L,NORB,NOB,1,ISYM)                                      MAIN
C    HINZE IS THE ROUTINE EMPLOYING THE HINAE-ROOTHAAN METHOD             MAIN
C    DIAGO EMPLOYS NORMAL DIAGONALIZATION                                 MAIN
      CALL HINZE(EXPCOE,ISYM,ORB,COMPL)                                   MAIN
C       CALL DIAGO(EXPCOE,NOBT,ISYM,FH1,FH2,FHH1,FHH2,FHH3,FHH4,WK)       MAIN
      CALL TIME(1,1)                                                      MAIN
      WRITE(8,902)COMPL                                                   MAIN
      COMPLV(ISYM)=COMPL                                                  MAIN
902   FORMAT('0  COMPL= ',1PD8.1)                                         MAIN
      CALL RENORM(NOB,NORB,ISYM,EXPCOE,S)                                 MAIN
      CALL ENER(ISYM,EXPCOE,FH1,FH2,NOBT,ORB,ENERGY)                      MAIN
      CALL EXVAHH(FHH1,FHH2,FHH3,FHH4,EXPCOE,ISYM,NOBT,ORB,EXHH)          MAIN
      CALL OUT01(EXPCOE,NOB,NORB,ISYM)                                    MAIN
21    CONTINUE                                                            MAIN
      CALL OUTPUT(EXPCOE,ORBEXP,EXHH,ENERGY,WK,COMPL,ORB,NOBT,METHOD,ITEMAIN
     .R,QN,ICOMPL,CHARGE)                                                 MAIN
      CALL CNVRGC(EXPCOE,ITER,NOBT,ORB,&22)                               MAIN
      IF(COMPL.LT.1.D-13)GOTO22                                           MAIN
      IF(ITER.EQ.1)CALL OUT5(INTLI2,INTNO2,NULL2,INTLI3,INTNO3,NULL3)     MAIN
      CALL AITKEN(EXPCOE,ITER,NOBT,ORB)                                   MAIN
20    CALL REWIND(3)                                                      MAIN
      IF(IOPT.EQ.0)GOTO23                                                 MAIN
      IF(IOPT.EQ.0)READ(5,903)III
```

```
903   FORMAT(20I4)                                                     MAIN
      GOTO12                                                           MAIN
22    CALL REWIND(3)                                                   MAIN
      ICOMPL=ICOMPL+1                                                  MAIN
      IF(ICOMPL.EQ.1)GOTO5                                             MAIN
23    IOPT=IOPT+1                                                      MAIN
      IF(IOPT.GT.1)GOTO12                                              MAIN
      CALL OPTIM(ORB,ENERGY,EXHH,WK,METHOD,IOPT,LIM4,TAU)              MAIN
      GOTO11                                                           MAIN
100   WRITE(6,901)                                                     MAIN
      STOP                                                             MAIN
900   FORMAT('1 ITERATION NO. ',I3)                                    MAIN
901   FORMAT(' LOGIOU HAS WRONG RETURN')                               MAIN
      DEBUG UNIT(9),SUBCHK,TRACE                                       MAIN
      END                                                              MAIN
      SUBROUTINE INPUT(ORB,NOBT,LIM1,LIM2,LIM3,LIM4,METHOD,INTLI2,INTLI3INPT
     .,QN,CHARGE,WK,CLOSED,ORBEXP,EXPCOE,INT1,FAC1,INT2,FAC2,INT3,FAC3,IINPT
     .NT4,FAC4,INTNO2,INTNO3,NULL2,NULL3,ITFA,IOPT,THRH,TAU)           INPT
      IMPLICIT REAL*8(A-H,O-Z)                                         INPT
      REAL*8 CHARGE,WK,ORBEXP(15),EXPCOE(5,10),FAC1(50),FAC2(100),FAC3(2INPT
     .00),FAC4(300)                                                    INPT
      INTEGER ORB(3),NOBT(3),QN(15),INT1(50),INT2(4,100),INT3(6,200),INTINPT
     .4(8,300),INTNO2(100),INTNO3(3,100)                               INPT
      LOGICAL CLOSED(3,4),NULL2(100),NULL3(100),NULL4(100)             INPT
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                          INPT
      READ(5,908,END=230)ORB,NOBT,ITFA,IOPT,IREAD,METHOD,INTLI2,INTLI3,NINPT
     .UM,IDEN                                                          INPT
      TAU=0.D0                                                         INPT
      IF(METHOD.EQ.6)TAU=DFLOAT(NUM)/DFLOAT(IDEN)                      INPT
      READ(5,908)QN                                                    INPT
      READ(5,909)CHARGE,WK,THRH                                        INPT
      READ(5,913)CLOSED                                                INPT
      READ(5,909)ORBEXP                                                INPT
      READ(5,908)NORBT,ISTA,(INNO(JA),JA=1,NORBT),(INNOR(JB),JB=1,NORBT)INPT
      DO 1 JA=1,NORBT                                                  INPT
      READ(5,909)(EXPCOE(JB,INNO(JA)),JB=1,5)                          INPT
1     CONTINUE                                                         INPT
      READ(5,908)LIM1,LIM2,LIM3,LIM4                                   INPT
      DO 5 JA=1,LIM1                                                   INPT
5     READ(5,900)INT1(JA),FAC1(JA)                                     INPT
      DO 6 JA=1,LIM2                                                   INPT
6     READ(5,910)(INT2(JB,JA),JB=1,4),FAC2(JA)                         INPT
      IF(LIM3.EQ.0)GOTO81                                              INPT
      DO 7 JA=1,LIM3                                                   INPT
7     READ(5,911)(INT3(JB,JA),JB=1,6),FAC3(JA)                         INPT
      IF(LIM4.EQ.0)GOTO81                                              INPT
      DO 8 JA=1,LIM4                                                   INPT
8     READ(5,912)(INT4(JB,JA),JB=1,8),FAC4(JA)                         INPT
81    IF(IREAD.EQ.0)RETURN                                             INPT
      WRITE(10,902)                                                    INPT
      WRITE(10,903)ORB,NOBT,ISTA,LIM1,LIM2,LIM3,LIM4                   INPT
      WRITE(10,914)INNO,INNOR                                          INPT
      WRITE(10,909)CHARGE,WK,THRH                                      INPT
      WRITE(10,914)ITFA,IREAD,METHOD,INTLI2,INTLI3                     INPT
      WRITE(10,915)QN                                                  INPT
      WRITE(10,909)ORBEXP                                              INPT
      DO 82 JA=1,3                                                     INPT
      NO=ORB(JA)                                                       INPT
      IF(NO.EQ.0)GOTO82                                                INPT
      WRITE(10,909)((EXPCOE(JB,INNO(ISTA(JA)+JC)),JB=1,5),JC=1,NO)     INPT
```

```
32    CONTINUE
      DO 120 JA=1,LIM1                                                         INPT
120   WRITE(10,904)FAC1(JA),INT1(JA)                                           INPT
      DO 12 JA=1,LIM2                                                          INPT
12    WRITE(10,904)FAC2(JA),(INT2(JB,JA),JB=1,4)                               INPT
      IF(METHOD.EQ.1.OR.LIM3.EQ.0)GOTO140                                      INPT
      WRITE(10,905)                                                            INPT
      DO 13 JA=1,LIM3                                                          INPT
13    WRITE(10,904)FAC3(JA),(INT3(JB,JA),JB=1,6)                               INPT
      WRITE(10,905)                                                            INPT
      DO 14 JA=1,LIM4                                                          INPT
14    WRITE(10,904)FAC4(JA),(INT4(JB,JA),JB=1,8)                               INPT
140   IF(IREAD.EQ.1)RETURN                                                     INPT
      READ(1'15000000,908)INTLI2,INTLI3                                        INPT
      DO 15 JA=1,INTLI2                                                        INPT
15    READ(1'(20000+JA)*1000,901)NULL2(JA),INTNO2(JA)                          INPT
      IF(IREAD.EQ.2)RETURN                                                     INPT
      DO 16 JA=1,INTLI3                                                        INPT
16    READ(1'(30000+JA)*1000,901)NULL3(JA),(INTNO3(JB,JA),JB=1,3)              INPT
      RETURN                                                                   INPT
230   STOP                                                                     INPT
900   FORMAT(33X,I3,3D25.15)                                                   INPT
901   FORMAT(L4,4I4)                                                           INPT
902   FORMAT('1')                                                              INPT
903   FORMAT('0',3(3I3,3X),4I3)                                                INPT
904   FORMAT(' ',D20.10,10X,8I4)                                               INPT
905   FORMAT(///)                                                              INPT
908   FORMAT(20I4)                                                             INPT
909   FORMAT(5D15.7)                                                           INPT
910   FORMAT(18X,2(3X,2I3),3D25.15)                                            INPT
911   FORMAT(9X,3(3X,2I3),3D25.15)                                             INPT
912   FORMAT(4(3X,2I3),3D25.15)                                                INPT
913   FORMAT(40L2)                                                             INPT
914   FORMAT('0',2(10I3,5X))                                                   INPT
915   FORMAT('0',20I4)                                                         INPT
      END                                                                      INPT
      SUBROUTINE OUT0(H,HH,S,EXPCOE,NOBT,ORB)                                  OUT1
      REAL*8 EXPCOE(5,10),H(5,5,3),HH(5,5,3),S(5,5,3),FH1(5,5,4),FHH1(5,OUT1
     .5,4),FH2(5,5,4),FHH2(5,5,4),FHH3(5,5,4),FHH4(5,5,4),LH2(5,5,4,4),LOUT1
     .HH2(5,5,4,4),LHH3(5,5,4,4),LHH4(5,5,4,4)                                 OUT1
      INTEGER ORB(3),NOBT(3),INTNO2(100),INTNO3(3,100)                         OUT1
      INTEGER I2OLD/0/,I3OLD/0/,I4OLD/0/                                       OUT1
      LOGICAL NULL2(100),NULL3(100)                                            OUT1
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                                  OUT1
      WRITE(6,905)                                                             OUT1
      DO 10 IS=1,3                                                             OUT1
      IF(ORB(IS).EQ.0)GOTO10                                                   OUT1
      NOB=NOBT(IS)                                                             OUT1
      WRITE(10,905)                                                            OUT1
      DO 11 JA=1,NOB                                                           OUT1
11    WRITE(10,906)(S(JA,JB,IS),JB=1,NOB)                                      OUT1
      WRITE(10,905)                                                            OUT1
      DO 20 JA=1,NOB                                                           OUT1
20    WRITE(10,906)(H(JA,JB,IS),JB=1,NOB)                                      OUT1
      WRITE(10,905)                                                            OUT1
      DO 30 JA=1,NOB                                                           OUT1
30    WRITE(10,906)(HH(JA,JB,IS),JB=1,NOB)                                     OUT1
10    CONTINUE                                                                 OUT1
      RETURN                                                                   OUT1
      ENTRY OUT01(EXPCOE,NOB,NORB,ISYM)                                        OUT1
```

```
      WRITE(6,905)                                                      OUT1
      IST=ISTA(ISYM)                                                    OUT1
      DO 40 JA=1,NORB                                                   OUT1
40    WRITE(6,907)(EXPCOE(JB,INNO(IST+JA)),JB=1,NOB)                    OUT1
      RETURN                                                            OUT1
      ENTRY OUT1(FH1,FHH1,NORB,NOB,ISYM)                                OUT1
      IST=ISTA(ISYM)                                                    OUT1
      WRITE(10,905)                                                     OUT1
      DO 50 JA=1,NORB                                                   OUT1
      WRITE(10,905)                                                     OUT1
      DO 50 JB=1,NOB                                                    OUT1
50    WRITE(10,906)(FH1(JB,JC,INNOR(IST+JA)),JC=1,NOB)                  OUT1
      WRITE(10,905)                                                     OUT1
      DO 60 JA=1,NORB                                                   OUT1
      WRITE(10,905)                                                     OUT1
      DO 60 JB=1,NOB                                                    OUT1
60    WRITE(10,906)(FHH1(JB,JC,JA),JC=1,NOB)                            OUT1
      RETURN                                                            OUT1
      ENTRY OUT2(FH2,FHH2,LH2,LHH2,NORB,NOB,NOL,NOF2,ISYM)              OUT1
      IST=ISTA(ISYM)                                                    OUT1
      DO 70 JA=1,NORB                                                   OUT1
      WRITE(10,905)                                                     OUT1
      DO 70 JB=1,NOB                                                    OUT1
70    WRITE(10,906)(FH2(JB,JC,INNOR(IST+JA)),JC=1,NOB)                  OUT1
      IF(NOF2.EQ.0)GOTO71                                               OUT1
      WRITE(10,905)                                                     OUT1
      DO 80 JA=1,NORB                                                   OUT1
      WRITE(10,905)                                                     OUT1
      DO 80 JB=1,NOB                                                    OUT1
71    IF(NOL.EQ.0)RETURN                                                OUT1
80    WRITE(10,906)(FHH2(JB,JC,INNOR(IST+JA)),JC=1,NOB)                 OUT1
      WRITE(10,905)                                                     OUT1
      DO 90 JA=1,NORB                                                   OUT1
      WRITE(10,905)                                                     OUT1
      DO 90 JB=1,NORB                                                   OUT1
      WRITE(10,905)                                                     OUT1
      DO 90 JC=1,NOB                                                    OUT1
90    WRITE(10,906)(LH2(JC,JD,INNOR(IST+JA),INNOR(IST+JB)),JD=1,NOB)    OUT1
      WRITE(10,905)                                                     OUT1
      DO 100 JA=1,NORB                                                  OUT1
      WRITE(10,905)                                                     OUT1
      DO 100 JB=1,NORB                                                  OUT1
      WRITE(10,905)                                                     OUT1
      DO 100 JC=1,NOB                                                   OUT1
100   WRITE (10,906) (LHH2(JC,JD,INNOR(IST+JA),INNOR(IST+JB)),JD=1,NOB) OUT1
      RETURN                                                            OUT1
      ENTRY OUT3(FHH3,LHH3,NORB,NOB,ISYM)                               OUT1
      IST=ISTA(ISYM)                                                    OUT1
      DO 120 JA=1,NORB                                                  OUT1
      WRITE(10,905)                                                     OUT1
      DO 120 JB=1,NOB                                                   OUT1
120   WRITE(10,906)(FHH3(JB,JC,INNOR(IST+JA)),JC=1,NOB)                 OUT1
      WRITE(10,905)                                                     OUT1
      DO 130 JA=1,NORB                                                  OUT1
      WRITE(10,905)                                                     OUT1
      DO 130 JB=1,NORB                                                  OUT1
      WRITE(10,905)                                                     OUT1
      DO 130 JC=1,NOB                                                   OUT1
130   WRITE(10,906)(LHH3(JC,JD,INNOR(IST+JA),INNOR(IST+JB)),JD=1,NOB)   OUT1
      RETURN                                                            OUT1
```

```
      ENTRY OUT4(FHH4,LHH4,NORB,NOB,NOL4,ISYM)                          OUT1
      IST=ISTA(ISYM)                                                    OUT1
      WRITE(10,905)                                                     OUT1
      DO 150 JA=1,NORB                                                  OUT1
      WRITE(10,905)                                                     OUT1
      DO 150 JB=1,NOB                                                   OUT1
150   WRITE(10,906)(FHH4(JB,JC,INNOR(IST+JA)),JC=1,NOB)                 OUT1
      IF(NOL4.EQ.0)RETURN                                               OUT1
      WRITE(10,905)                                                     OUT1
      DO 160 JA=1,NORB                                                  OUT1
      WRITE(10,905)                                                     OUT1
      DO 160 JB=1,NORB                                                  OUT1
      WRITE(10,905)                                                     OUT1
      DO 160 JC=1,NOB                                                   OUT1
160   WRITE(10,906)(LHH4(JC,JD,INNOR(IST+JA),INNOR(IST+JB)),JD=1,NOB)   OUT1
      RETURN                                                            OUT1
      ENTRY OUT5(I2NEW,INTNO2,NULL2,I3NEW,INTNO3,NULL3)                 OUT1
      WRITE(1'15000000,900)I2NEW,I3NEW                                  OUT1
      IF(I2NEW.EQ.I2OLD)GOTO190                                         OUT1
      IST=I2OLD+1                                                       OUT1
      I2OLD=I2NEW                                                       OUT1
      DO 180 JA=IST,I2NEW                                               OUT1
180   WRITE(1'(20000+JA)*1000,901)NULL2(JA),INTNO2(JA)                  OUT1
190   IF(I3NEW.EQ.I3OLD)RETURN                                          OUT1
      IST=I3OLD+1                                                       OUT1
      I3OLD=I3NEW                                                       OUT1
      DO 200 JA=IST,I3NEW                                               OUT1
200   WRITE(1'(30000+JA)*1000,901)NULL3(JA),(INTNO3(JB,JA),JB=1,3)      OUT1
      RETURN                                                            OUT1
900   FORMAT(20I4)                                                      OUT1
901   FORMAT(L4,4I4)                                                    OUT1
905   FORMAT(///)                                                       OUT1
906   FORMAT(' ',5D20.10)                                               OUT1
907   FORMAT(5D15.7)                                                    OUT1
      END                                                               OUT1
      SUBROUTINE ONEINT(HH,S)                                           ONEI
      IMPLICIT REAL*8(A-H,O-Z)                                          ONEI
      COMMON/ALL/EXPCOE,ORBEXP,H,CHARGE,QN,NOBT,ISYM,FDUB               ONEI
      COMMON/PROPER/SRM1(5,5,3),SRP1(5,5,3),SRP2(5,5,3)                 ONEI
      INTEGER QN(15),ISTA(3)/0,5,10/,FDUB,NOBT(3),IC(3)/'  S-','  P-',' ONEI
     . D-'/                                                             ONEI
      REAL*8 EXPCOE(5,10),ORBEXP(15),H(5,5,3),HH(5,5,3),S(5,5,3),VEC(5) ONEI
     .,MAT(25)                                                          ONEI
      DO 1 JA=1,3                                                       ONEI
      L=JA-1                                                            ONEI
      LIM=NOBT(JA)                                                      ONEI
      IF(LIM.EQ.0)GOTO1                                                 ONEI
      DO 2 JB=1,LIM                                                     ONEI
      N1B=QN(ISTA(JA)+JB)                                               ONEI
      OE1B=ORBEXP(ISTA(JA)+JB)                                          ONEI
      EN1B=ENMI(N1B,L,0,OE1B)                                           ONEI
      DO 2 JC=1,LIM                                                     ONEI
      N1K=QN(ISTA(JA)+JC)                                               ONEI
      OE1K=ORBEXP(ISTA(JA)+JC)                                          ONEI
      EN1K=EN1B*ENMI(N1K,L,0,OE1K)                                      ONEI
      CALL ONEI(N1B,L,0,OE1B,N1K,L,0,OE1K,CHARGE,SE,HE,HHE,RM1,RP1,RP2) ONEI
      SRM1(JB,JC,JA)=RM1*EN1K                                           ONEI
      SRP1(JB,JC,JA)=RP1*EN1K                                           ONEI
      SRP2(JB,JC,JA)=RP2*EN1K                                           ONEI
      S(JB,JC,JA)=SE*EN1K
```

```
      H(JB,JC,JA)=HE*EN1K                                               ONEI
2     HH(JB,JC,JA)=HHE*EN1K                                             ONEI
1     CONTINUE                                                          ONEI
      RETURN                                                            ONEI
C   COMPUTE THE VALUE OF THE DETERMINANT OF THE S-MATRICES              ONEI
999   DO 3 JA=1,3                                                       ONEI
      NOB=NOBT(JA)                                                      ONEI
      IF(NOB.EQ.0)GOTO3                                                 ONEI
C   FILL UP THE MATRIX FOR USE IN GAUSS                                 ONEI
      DO 4 JB=1,NOB                                                     ONEI
      VEC(JB)=DFLOAT(JB)                                                ONEI
      IND=(JB-1)*NOB                                                    ONEI
      DO 4 JC=1,NOB                                                     ONEI
4     MAT(IND+JB)=S(JB,JC,JA)                                           ONEI
      CALL GAUSS(MAT,VEC,NOB)                                           ONEI
C   COMPUTE THE DETERMINANT                                             ONEI
      DET=1.D0                                                          ONEI
      DO 5 JB=1,NOB                                                     ONEI
5     DET=DET*MAT((JB-1)*NOB+JB)                                        ONEI
      WRITE(6,900)IC(JA),DET                                            ONEI
3     CONTINUE                                                          ONEI
      RETURN                                                            ONEI
900   FORMAT(' THE VALUE OF THE DETERMINANT OF THE S-MATRIX FOR',A4,'OR ONEI
     .BITAL IS',1PD12.3)                                                ONEI
      END                                                               ONEI
      SUBROUTINE RENORM(NOB,NORB,ISYM,EXPCOE,S)                         NORM
      REAL*8 EXPCOE(5,10),S(5,5,3),SV(10),EM(10),R1(4,5)                NORM
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                           NORM
      IST=ISTA(ISYM)                                                    NORM
C   MULTIPLY CMAT*S                                                     NORM
      DO 1 JA=1,NORB                                                    NORM
      DO 1 JB=1,NOB                                                     NORM
      R1(JA,JB)=0.D0                                                    NORM
      DO 1 JC=1,NOB                                                     NORM
1     R1(JA,JB)=R1(JA,JB)+EXPCOE(JC,INNO(IST+JA))*S(JC,JB,ISYM)         NORM
C MULTIPLY R1*CAMT                                                      NORM
      DO 2 JA=1,NORB                                                    NORM
      DO 2 JB=1,JA                                                      NORM
      IJ=JB+JA*(JA-1)/2                                                 NORM
      SV(IJ)=0.D0                                                       NORM
      DO 2 JC=1,NOB                                                     NORM
2     SV(IJ)=SV(IJ)+R1(JA,JC)*EXPCOE(JC,INNO(IST+JB))                   NORM
C   RENORMALIZE SV                                                      NORM
      CALL SOMS(NORB,SV,EM)                                             NORM
C   COMPUTE THE RENORMALIZED STARTING VECTORS                           NORM
C   MULTIPLY EM*CMAT                                                    NORM
      DO 3 JA=1,NORB                                                    NORM
      DO 3 JB=1,NOB                                                     NORM
      R1(JA,JB)=0.D0                                                    NORM
      DO 3 JC=1,JA                                                      NORM
3     R1(JA,JB)=R1(JA,JB)+EM(JC+JA*(JA-1)/2)*EXPCOE(JB,INNO(IST+JC))    NORM
C   PUT R1 INTO EXPCOE                                                  NORM
      DO 4 JA=1,NORB                                                    NORM
      DO 4 JB=1,NOB                                                     NORM
4     EXPCOE(JB,INNO(IST+JA))=R1(JA,JB)                                 NORM
      RETURN                                                            NORM
      END                                                               NORM
      SUBROUTINE ONEEL(NOB,ISYM,H,HH)                                   ONEE
C   SETS UP THE ONEELECTRON MATRICES                                    ONEE
      IMPLICIT REAL*8(A-H,O-Z)                                          ONEE
```

```
      REAL*8 H(5,5,3),HH(5,5,3)                                         ONEE
      COMMON/ONE/FAC1(50),FHH1(5,5,4),FH1(5,5,4),INT1(50),LIM1          ONEE
      INTEGER SYMCHE,ORB(3),NOBT(3)                                     ONEE
      DO 2 JA=1,LIM1                                                    ONEE
      I1B=INT1(JA)                                                      ONEE
      IF(SYMCHE(I1B).NE.ISYM)GOTO2                                      ONEE
      GOTO(4,5,4,4,4,6,5,5,5,4,4,4,4,4,7,6,6,6,5,5,5,5,5),I1B           ONEE
    4 JM1=1                                                             ONEE
      GOTO8                                                             ONEE
    5 JM1=2                                                             ONEE
      GOTO8                                                             ONEE
    6 JM1=3                                                             ONEE
      GOTO8                                                             ONEE
    7 JM1=4                                                             ONEE
    8 DO 3 JB=1,NOB                                                     ONEE
      DO 3 JC=1,NOB                                                     ONEE
      FH1(JB,JC,JM1)=FH1(JB,JC,JM1)+H(JB,JC,ISYM)*FAC1(JA)              ONEE
    3 FHH1(JB,JC,JM1)=FHH1(JB,JC,JM1)+HH(JB,JC,ISYM)*FAC1(JA)           ONEE
    2 CONTINUE                                                          ONEE
      RETURN                                                            ONEE
      END                                                               ONEE
      SUBROUTINE TWOELE                                                 TWOE
      IMPLICIT REAL*8(A-H,O-Z)                                          TWOE
      COMMON/SYM/IDAR(8,10)                                             TWOE
      COMMON/ALL/EXPCOE,ORBEXP,H,CHARGE,QN,NOBT,ISYM,FDUB               TWOE
      COMMON/SPLI1/I1,I2,J1,J2,K1,K2,L1,L2,LIMI,LIMJ,LIMK,LIML,JMI,JMJ,JTWOE
     .MK,JML,IEXP,JEXP,KEXP,LEXP                                        TWOE
      COMMON/TWO/FH2,FHH2,FAC2,LH2,LHH2,INT2,INTNO2,NULL2,LIM2,INTLI2   TWOE
      COMMON/INTRA2/INTEG1,INTEG2                                       TWOE
      REAL*8 INTEG1(5,5,5,5),INTEG2(5,5,5,5),FH2(5,5,4),FHH2(5,5,4),FAC2TWOE
     .(100),H(5,5,3),EXPCOE(5,10),ORBEXP(15),LH2(5,5,4,4),LHH2(5,5,4,4) TWOE
      INTEGER SYMCHE,INT2(4,100),IV(4),IM2(4,2)/1,2,3,4,3,4,1,2/,NOBT(3)TWOE
     .,INTNO2(100),QN(15),FDUB                                          TWOE
      LOGICAL LS1,NULL2(100),RC,CHL1                                    TWOE
      DO 1 JA=1,LIM2                                                    TWOE
      CHL1=.FALSE.                                                      TWOE
      DO 2 JB=1,2                                                       TWOE
      I1=INT2(IM2(1,JB),JA)                                             TWOE
      IF(SYMCHE(I1).NE.ISYM)GOTO2                                       TWOE
      I2=INT2(IM2(2,JB),JA)                                             TWOE
      J1=INT2(IM2(3,JB),JA)                                             TWOE
      J2=INT2(IM2(4,JB),JA)                                             TWOE
      CALL SPLIT2(NOBT,ISY1B,ISY2B)                                     TWOE
      LS1=ISY1B.NE.ISY2B                                                TWOE
      IF(CHL1)GOTO6                                                     TWOE
      CHL1=.TRUE.                                                       TWOE
      CALL TWINT(1,1,2,3,4,RC)                                          TWOE
      IF(RC)GOTO1                                                       TWOE
    6 DO 3 JE=1,LIMJ                                                    TWOE
      IV(IDAR(3,1))=JE                                                  TWOE
      EXPE=EXPCOE(JE,JEXP)*FAC2(JA)                                     TWOE
      DO 3 JF=1,LIMJ                                                    TWOE
      IV(IDAR(4,1))=JF                                                  TWOE
      EXP=EXPE*EXPCOE(JF,JEXP)                                          TWOE
      EXPL1=FAC2(JA)*EXPCOE(JF,JEXP)                                    TWOE
      DO 3 JC=1,LIMI                                                    TWOE
      IV(IDAR(1,1))=JC                                                  TWOE
      DO 3 JD=1,LIMI                                                    TWOE
      IV(IDAR(2,1))=JD                                                  TWOE
      EXPL=EXPL1*EXPCOE(JD,IEXP)
```

```
      X1=INTEG1(IV(1),IV(2),IV(3),IV(4))                                TWOE
      FH2(JC,JD,JMI)=FH2(JC,JD,JMI)+EXP*X1                              TWOE
      X2=INTEG2(IV(1),IV(2),IV(3),IV(4))                                TWOE
      FHH2(JC,JD,JMI)=FHH2(JC,JD,JMI)+EXP*X2                            TWOE
      IF(LS1)GOTO3                                                      TWOE
      LH2(JC,JE,JMI,JMJ)=LH2(JC,JE,JMI,JMJ)+EXPL*X1                     TWOE
      LHH2(JC,JE,JMI,JMJ)=LHH2(JC,JE,JMI,JMJ)+EXPL*X2                   TWOE
3     CONTINUE                                                          TWOE
2     CONTINUE                                                          TWOE
1     CONTINUE                                                          TWOE
      RETURN                                                            TWOE
      END                                                               TWOE
      SUBROUTINE TWINT(INDEX,I1,I2,I3,I4,RC)                            TWIN
      IMPLICIT REAL*8(A-H,O-Z)                                          TWIN
      COMMON/SYM/IDAR(8,10)                                             TWIN
      COMMON/INTRA2/INTEG1,INTEG2                                       TWIN
      COMMON/ALL/EXPCOE,ORBEXP,H,CHARGE,QN,NOBT,ISYM,FDUB               TWIN
      COMMON/TWO/FH2,FHH2,FAC2,LH2,LHH2,INT2,INTNO2,NULL2,LIM2,INTLI2   TWIN
      REAL*8 INTEG1(5,5,5,5),INTEG2(5,5,5,5),ORBEXP(15),EXPCOE(5,10),H(5TWIN
     .,5,3),FAC2(100),FH2(5,5,4),FHH2(5,5,4),LH2(5,5,4,4),LHH2(5,5,4,4) TWIN
      INTEGER QN(15),ISTA(3)/0,5,10/,NOBT(3),FDUB,INFO(4),INTNO2(100),INTWIN
     .T2(4,100)                                                         TWIN
      LOGICAL LSYM,NULL2(100),RC                                        TWIN
      INTEGER*2 LEN/5000/                                               TWIN
      NULI1=0                                                           TWIN
      IF(INDEX.GT.1)GOTO6                                               TWIN
      CALL SYMAS3(NOBT,4)                                               TWIN
      IF(INTLI2.EQ.0)GOTO2                                              TWIN
      DO 1 JA=1,INTLI2                                                  TWIN
      IF(INTNO2(JA).EQ.IDAR(1,8))GOTO3                                  TWIN
1     CONTINUE                                                          TWIN
2     INTLI2=INTLI2+1                                                   TWIN
      IF(INTLI2.LE.100)GOTO4                                            TWIN
      WRITE(6,900)                                                      TWIN
900   FORMAT(' DIMENSION OF INTNO2 EXCEEDED')                           TWIN
      STOP                                                              TWIN
4     INTNO2(INTLI2)=IDAR(1,8)                                          TWIN
6     LIM1B=IDAR(I1,4)                                                  TWIN
      L1B=IDAR(I1,5)                                                    TWIN
      M1B=IDAR(I1,6)                                                    TWIN
      I1B=ISTA(IDAR(I1,7))                                              TWIN
      LIM1K=IDAR(I2,4)                                                  TWIN
      L1K=IDAR(I2,5)                                                    TWIN
      M1K=IDAR(I2,6)                                                    TWIN
      I1K=ISTA(IDAR(I2,7))                                              TWIN
      LIM2B=IDAR(I3,4)                                                  TWIN
      L2B=IDAR(I3,5)                                                    TWIN
      M2B=IDAR(I3,6)                                                    TWIN
      I2B=ISTA(IDAR(I3,7))                                              TWIN
      LIM2K=IDAR(I4,4)                                                  TWIN
      L2K=IDAR(I4,5)                                                    TWIN
      M2K=IDAR(I4,6)                                                    TWIN
      I2K=ISTA(IDAR(I4,7))                                              TWIN
      F1=1.D0                                                           TWIN
      F2=1.D0                                                           TWIN
      IF(IDAR(I1,2).NE.IDAR(I2,2))F1=0.D0                               TWIN
      IF(IDAR(I3,2).NE.IDAR(I4,2))F2=0.D0                               TWIN
      DO 5 JA=1,LIM1B                                                   TWIN
      N1B=QN(I1B+JA)                                                    TWIN
      OE1B=ORBEXP(I1B+JA)
```

```
      EN1B=ENMI(N1B,L1B,M1B,OE1B)                                        TWIN
      DO 5 JB=1,LIM1K                                                    TWIN
      N1K=QN(I1K+JB)                                                     TWIN
      OE1K=ORBEXP(I1K+JB)                                                TWIN
      EN1K=EN1B*ENMI(N1K,L1K,M1K,OE1K)                                   TWIN
      H1=F1*H(JA,JB,IDAR(I1,7))                                          TWIN
      DO 5 JC=1,LIM2B                                                    TWIN
      N2B=QN(I2B+JC)                                                     TWIN
      OE2B=ORBEXP(I2B+JC)                                                TWIN
      EN2B=EN1K*ENMI(N2B,L2B,M2B,OE2B)                                   TWIN
      DO 5 JD=1,LIM2K                                                    TWIN
      N2K=QN(I2K+JD)                                                     TWIN
      OE2K=ORBEXP(I2K+JD)                                                TWIN
      EN2K=EN2B*ENMI(N2K,L2K,M2K,OE2K)                                   TWIN
      H2=F2*H(JC,JD,IDAR(I3,7))                                          TWIN
      X1=EN2K*REPI(1,N1B,L1B,M1B,OE1B,N2B,L2B,M2B,OE2B,N1K,L1K,M1K,OE1K,TWIN
     .N2K,L2K,M2K,OE2K,1,0,0,1.D0,1,0,0,1.D0)                            TWIN
      X2=REPI(2,N1B,L1B,M1B,OE1B,N2B,L2B,M2B,OE2B,N1K,L1K,M1K,OE1K,N2K,LTWIN
     .2K,M2K,OE2K,1,0,0,1.D0,1,0,0,1.D0)                                 TWIN
      CALL HR(N1B,L1B,M1B,OE1B,N2B,L2B,M2B,OE2B,N1K,L1K,M1K,OE1K,N2K,L2KTWIN
     .,M2K,OE2K,CHARGE,X3,X4)                                            TWIN
      INTEG1(JA,JB,JC,JD)=X1                                             TWIN
      X7=2.D0*H1*H2+(X3+X4+X2)*EN2K                                      TWIN
      INTEG2(JA,JB,JC,JD)=X7                                             TWIN
5     IF(X1.EQ.0.D0)NULI1=NULI1+1                                        TWIN
      MULT=LIM1B*LIM1K*LIM2B*LIM2K                                       TWIN
      IF(NULI1.LT.MULT)GOTO7                                             TWIN
      NULL2(INTLI2)=.TRUE.                                               TWIN
      RC=.TRUE.                                                          TWIN
      RETURN                                                             TWIN
7     RC=.FALSE.                                                         TWIN
      NULL2(INTLI2)=.FALSE.                                              TWIN
      CALL NOTE(FDUB,INFO)                                               TWIN
      WRITE(1'(2000+INTLI2)*1000)INFO(2),INFO(2),INFO(3),INFO(4)         TWIN
      CALL WRITE(INTEG1,LEN,0,LNR,2,&100)                                TWIN
      CALL NOTE(FDUB,INFO)                                               TWIN
      WRITE(1'(2500+INTLI2)*1000)INFO(2),INFO(2),INFO(3),INFO(4)         TWIN
      CALL WRITE(INTEG2,LEN,0,LNR,2,&100)                                TWIN
      RETURN                                                             TWIN
3     RC=NULL2(JA)                                                       TWIN
      IF(RC)RETURN                                                       TWIN
      READ(1'(2000+JA)*1000)INFO                                         TWIN
      CALL POINT(FDUB,INFO,1)                                            TWIN
      CALL READ(INTEG1,LEN,0,LNR,2,&100)                                 TWIN
      READ(1'(2500+JA)*1000)INFO                                         TWIN
      CALL POINT(FDUB,INFO,1)                                            TWIN
      CALL READ(INTEG2,LEN,0,LNR,2,&100)                                 TWIN
      RETURN                                                             TWIN
100   WRITE(6,901)                                                       TWIN
      STOP                                                               TWIN
901   FORMAT(' WRONG RETURN IN I/O ROUT')                                TWIN
      END                                                                THRE
      SUBROUTINE THREEL(LIMDI3,NORB,NOB)                                 THRE
      IMPLICIT REAL*8(A-H,O-Z)                                           THRE
      COMMON/SYM/IDAR(8,10)                                              THRE
      COMMON/THREE/FAC3(200),FHH3(5,5,4),LHH3(5,5,4,4),INT3(6,200),INTNOTHRE
     .3(3,100),LIM3,INTLI3,NULL3(100)                                    THRE
      COMMON/DENSI3/DIJ(15,15),DIK(15,15),DJK(15,15),CDIJK(5,5,15),CDIKJTHRE
     .(5,5,15),CDJKI(5,5,15)                                             THRE
      COMMON/ALL/EXPCOE,ORBEXP,H,CHARGE,QN,NOBT,ISYM,FDUB
```

```
      COMMON/INTRA3/INTEG                                                       THRE
      COMMON/SPLI1/I1,I2,J1,J2,K1,K2,L1,L2,LIMI,LIMJ,LIMK,LIML,JMI,JMJ,JTHRE
     .MK,JML,IEXP,JEXP,KEXP,LEXP                                                THRE
      LOGICAL NULL3,LOI,LOJ,LOK,WDH/.FALSE./                                    THRE
      INTEGER QN(15),NOBT(3),FDUB,IV(6),SYMCHE                                  THRE
      REAL*8 INTEG(5,5,5,5,5,5),ORBEXP(15),EXPCOE(5,10),LHH3,H(5,5,3)           THRE
      IJN(I,J)=MIN0(I,J)+(MAX0(I,J)*(MAX0(I,J)-1))/2                            THRE
      IF(WDH)GOTO14                                                             THRE
      WDH=.TRUE.                                                                THRE
      DO 13 JA=1,100                                                            THRE
13    NULL3(JA)=.FALSE.                                                         THRE
14    DO 1 JA=1,LIM3                                                            THRE
      IF(NULL3(JA))GOTO1                                                        THRE
      I1=INT3(1,JA)                                                             THRE
      I2=INT3(2,JA)                                                             THRE
      J1=INT3(3,JA)                                                             THRE
      J2=INT3(4,JA)                                                             THRE
      K1=INT3(5,JA)                                                             THRE
      K2=INT3(6,JA)                                                             THRE
      CALL SYM34(NOBT,LOI,LOJ,LOK,LOK,ISYM,&1,3)                                THRE
      IF(INTLI3.EQ.0)GOTO3                                                      THRE
      DO 2 JB=1,INTLI3                                                          THRE
      IF(INTNO3(1,JB).NE.IDAR(1,3))GOTO2                                        THRE
      IF(INTNO3(2,JB).NE.IDAR(3,3))GOTO2                                        THRE
      IF(INTNO3(3,JB).NE.IDAR(5,3))GOTO2                                        THRE
      READ(1'(3000+JB)*1000)INFO                                                THRE
      CALL POINT(FDUB,INFO,1)                                                   THRE
      READ(2)INTEG                                                              THRE
      GOTO5                                                                     THRE
2     CONTINUE                                                                  THRE
3     INTLI3=INTLI3+1                                                           THRE
      INTNO3(1,INTLI3)=IDAR(1,3)                                                THRE
      INTNO3(2,INTLI3)=IDAR(3,3)                                                THRE
      INTNO3(3,INTLI3)=IDAR(5,3)                                                THRE
      CALL TINT3(INTLI3,NULL3,RC)                                               THRE
      IF(NULL3(INTLI3))GOTO1                                                    THRE
5     CALL DENS3(LIMDI3,EXPCOE)                                                 THRE
      ASSIGN 8 TO ICASE                                                         THRE
      IF(LOJ)ASSIGN 7 TO ICASE                                                  THRE
      IF(LOI)ASSIGN 6 TO ICASE                                                  THRE
      DO 4 IB=1,LIMI                                                            THRE
      IV(1)=IB                                                                  THRE
      DO 4 IK=1,LIMI                                                            THRE
      IV(2)=IK                                                                  THRE
      IBK=IJN(IB,IK)                                                            THRE
      DO 4 JB=1,LIMJ                                                            THRE
      IV(3)=JB                                                                  THRE
      DO 4 JK=1,LIMJ                                                            THRE
      IV(4)=JK                                                                  THRE
      JBK=IJN(JB,JK)                                                            THRE
      DO 4 KB=1,LIMK                                                            THRE
      IV(5)=KB                                                                  THRE
      DO 4 KK=1,LIMK                                                            THRE
      IV(6)=KK                                                                  THRE
      KBK=IJN(KB,KK)                                                            THRE
      X1=(INTEG(IV(IDAR(1,1)),IV(IDAR(2,1)),IV(IDAR(3,1)),IV(IDAR(4,1)),THRE
     .IV(IDAR(5,1)),IV(IDAR(6,1))))*FAC3(JA)                                    THRE
      GOTOICASE,(6,7,8)                                                         THRE
6     FHH3(IB,IK,JMI)=FHH3(IB,IK,JMI)+X1*DJK(JBK,KBK)                           THRE
      LHH3(IK,JB,JMI,JMJ)=LHH3(IK,JB,JMI,JMJ)+X1*CDIJK(IB,JK,KBK)               THRE
```

```
      LHH3(IK,JK,JMI,JMJ)=LHH3(IK,JK,JMI,JMJ)+X1*CDIJK(IB,JB,KBK)          THRE
      LHH3(IB,JK,JMI,JMJ)=LHH3(IB,JK,JMI,JMJ)+X1*CDIJK(IK,JB,KBK)          THRE
      LHH3(IB,JB,JMI,JMJ)=LHH3(IB,JB,JMI,JMJ)+X1*CDIJK(IK,JK,KBK)          THRE
      LHH3(IK,KB,JMI,JMK)=LHH3(IK,KB,JMI,JMK)+X1*CDIKJ(IB,KK,JBK)          THRE
      LHH3(IK,KK,JMI,JMK)=LHH3(IK,KK,JMI,JMK)+X1*CDIKJ(IB,KB,JBK)          THRE
      LHH3(IB,KB,JMI,JMK)=LHH3(IB,KB,JMI,JMK)+X1*CDIKJ(IK,KK,JBK)          THRE
      LHH3(IB,KK,JMI,JMK)=LHH3(IB,KK,JMI,JMK)+X1*CDIKJ(IK,KB,JBK)          THRE
7     FHH3(JB,JK,JMJ)=FHH3(JB,JK,JMJ)+X1*DIK(IBK,KBK)                      THRE
      LHH3(JK,KB,JMJ,JMK)=LHH3(JK,KB,JMJ,JMK)+X1*CDJKI(JB,KK,IBK)          THRE
      LHH3(JK,KK,JMJ,JMK)=LHH3(JK,KK,JMJ,JMK)+X1*CDJKI(JB,KB,IBK)          THRE
      LHH3(JB,KK,JMJ,JMK)=LHH3(JB,KK,JMJ,JMK)+X1*CDJKI(JK,KB,IBK)          THRE
      LHH3(JB,KB,JMJ,JMK)=LHH3(JB,KB,JMJ,JMK)+X1*CDJKI(JK,KK,IBK)          THRE
8     FHH3(KB,KK,JMK)=FHH3(KB,KK,JMK)+X1*DIJ(IBK,JBK)                      THRE
4     CONTINUE                                                             THRE
1     CONTINUE                                                             THRE
      DO 10 JA=1,NORB                                                      THRE
      DO 11 JB=1,NOB                                                       THRE
      DO 11 JC=1,JB                                                        THRE
      FHH3(JB,JC,JA)=0.5D0*(FHH3(JB,JC,JA)+FHH3(JC,JB,JA))                 THRE
      FHH3(JC,JB,JA)=FHH3(JB,JC,JA)                                        THRE
      LHH3(JB,JC,JA,JA)=(LHH3(JB,JC,JA,JA)+LHH3(JC,JB,JA,JA))*0.25D0       THRE
11    LHH3(JC,JB,JA,JA)=LHH3(JB,JC,JA,JA)                                  THRE
      IS=JA+1                                                              THRE
      IF(IS.GT.NORB)RETURN                                                 THRE
      DO 12 JB=IS,NORB                                                     THRE
      DO 12 JC=1,NOB                                                       THRE
      DO 12 JD=1,NOB                                                       THRE
      LHH3(JD,JC,JA,JB)=LHH3(JD,JC,JA,JB)*0.25D0                           THRE
12    LHH3(JC,JD,JB,JA)=LHH3(JD,JC,JA,JB)                                  THRE
10    CONTINUE                                                             THRE
      RETURN                                                               THRE
      END                                                                  THRE
      SUBROUTINE TINT3(INTLI3,NULL3,RC)                                    TINT
      IMPLICIT REAL*8(A-H,O-Z)                                             TINT
      REAL*8 EXPCOE(5,10),ORBEXP(15),INTEG(5,5,5,5,5,5),INTEG1(5,5,5,5),TINT
     .H(5,5,3),INTEG2(5,5,5,5),INTEG3(5,5,5,5),LH2,LHH2                    TINT
      INTEGER QN(15),NOBT(3),FDUB,INFO(4)                                  TINT
      COMMON/ALL/EXPCOE,ORBEXP,H,CHARGE,QN,NOBT,ISYM,FDUB                  TINT
      COMMON/INTRA3/INTEG                                                  TINT
      COMMON/SYM/IDAR(8,10)                                                TINT
      COMMON/TWO/FH2(5,5,4),FHH2(5,5,4),FAC2(100),LH2(5,5,4,4),LHH2(5,     TINT
     .5,4,4),INT2(4,100),INTNO2(100),NULL2(100),LIM2,INTLI2                TINT
      LOGICAL RC1,RC2,RC3,RC,NULL2,NULL3(100)                              TINT
      INTEGER*2 LEN1/5000/                                                 TINT
      NULI=0                                                               TINT
      DO 4 JA=2,6,2                                                        TINT
      DO 5 JB=1,INTLI2                                                     TINT
      IF(INTNO2(JB).NE.IDAR(JA,3))GOTO5                                    TINT
      IF(NULL2(JB))GOTO51                                                  TINT
      READ(1'(2000+JB)*1000)INFO                                           TINT
      CALL POINT(FDUB,INFO,1)                                              TINT
      IF(JA-4)6,7,8                                                        TINT
6     CALL READ(INTEG1,LEN1,0,LNR,2,&100)                                  TINT
      RC1=.FALSE.                                                          TINT
      GOTO4                                                                TINT
7     CALL READ(INTEG2,LEN1,0,LNR,2,&100)                                  TINT
      RC2=.FALSE.                                                          TINT
      GOTO4                                                                TINT
8     CALL READ(INTEG3,LEN1,0,LNR,2,&100)                                  TINT
      RC3=.FALSE.                                                          TINT
```

```
      GOTO4                                                   TINT
51    IF(JA-4)52,53,54                                        TINT
52    RC1=.TRUE.                                              TINT
      GOTO4                                                   TINT
53    RC2=.TRUE.                                              TINT
      GOTO4                                                   TINT
54    RC3=.TRUE.                                              TINT
      GOTO4                                                   TINT
5     CONTINUE                                                TINT
      INTLI2=INTLI2+1                                         TINT
      IF(INTLI2.LE.100)GOTO10                                 TINT
      WRITE(8,901)                                            TINT
901   FORMAT('  MORE THAN 100 2-EL-INTS IN TRINT')            TINT
      STOP                                                    TINT
10    INTNO2(INTLI2)=IDAR(JA,3)                               TINT
      IF(JA-4)11,12,13                                        TINT
11    CALL TWINT(3,3,4,5,6,RC1)                               TINT
      IF(RC1)GOTO4                                            TINT
      GOTO14                                                  TINT
12    CALL TWINT(3,1,2,5,6,RC2)                               TINT
      IF(RC2)GOTO4                                            TINT
      GOTO14                                                  TINT
13    CALL TWINT(3,1,2,3,4,RC3)                               TINT
      IF(RC3)GOTO4                                            TINT
14    READ(1'(2000+INTLI2)*1000)INFO                          TINT
      CALL POINT(FDUB,INFO,1)                                 TINT
      IF(JA-4)15,16,17                                        TINT
15    CALL READ(INTEG1,LEN1,0,LNR,2,&100)                     TINT
      GOTO4                                                   TINT
16    CALL READ(INTEG2,LEN1,0,LNR,2,&100)                     TINT
      GOTO4                                                   TZNT
17    CALL READ(INTEG3,LEN1,0,LNR,2,&100)                     TINT
4     CONTINUE                                                TINT
3     F1=2.D0                                                 TINT
      F2=2.D0                                                 TINT
      F3=2.D0                                                 TINT
      IF(IDAR(1,2).NE.IDAR(2,2).OR.RC1)F1=0.D0                TINT
      IF(IDAR(3,2).NE.IDAR(4,2).OR.RC2)F2=0.D0                TINT
      IF(IDAR(5,2).NE.IDAR(6,2).OR.RC3)F3=0.D0                TINT
      LIM1B=IDAR(1,4)                                         TINT
      L1B=IDAR(1,5)                                           TINT
      M1B=IDAR(1,6)                                           TINT
      IB1=(IDAR(1,7)-1)*5                                     TINT
      LIM1K=IDAR(2,4)                                         TINT
      L1K=IDAR(2,5)                                           TINT
      M1K=IDAR(2,6)                                           TINT
      IK1=(IDAR(2,7)-1)*5                                     TINT
      LIM2B=IDAR(3,4)                                         TINT
      L2B=IDAR(3,5)                                           TINT
      M2B=IDAR(3,6)                                           TINT
      IB2=(IDAR(3,7)-1)*5                                     TINT
      LIM2K=IDAR(4,4)                                         TINT
      L2K=IDAR(4,5)                                           TINT
      M2K=IDAR(4,6)                                           TINT
      IK2=(IDAR(4,7)-1)*5                                     TINT
      LIM3B=IDAR(5,4)                                         TINT
      L3B=IDAR(5,5)                                           TINT
      M3B=IDAR(5,6)                                           TINT
      IB3=(IDAR(5,7)-1)*5                                     TINT
      LIM3K=IDAR(6,4)                                         TINT
```

```
      L3K=IDAR(6,5)                                                     TINT
      M3K=IDAR(6,6)                                                     TINT
      IK3=(IDAR(6,7)-1)*5                                               TINT
      DO 1 JA=1,LIM1B                                                   TINT
      N1B=QN(IB1+JA)                                                    TINT
      OE1B=ORBEXP(IB1+JA)                                               TINT
      EN1B=ENMI(N1B,L1B,M1B,OE1B)                                       TINT
      DO 1 JB=1,LIM1K                                                   TINT
      N1K=QN(IK1+JB)                                                    TINT
      OE1K=ORBEXP(IK1+JB)                                               TINT
      EN1K=EN1B*ENMI(N1K,L1K,M1K,OE1K)                                  TINT
      H1=F1*H(JA,JB,IDAR(1,7))                                          TINT
      DO 1 JC=1,LIM2B                                                   TINT
      N2B=QN(IB2+JC)                                                    TINT
      OE2B=ORBEXP(IB2+JC)                                               TINT
      EN2B=EN1K*ENMI(N2B,L2B,M2B,OE2B)                                  TINT
      DO 1 JD=1,LIM2K                                                   TINT
      N2K=QN(IK2+JD)                                                    TINT
      OE2K=ORBEXP(IK2+JD)                                               TINT
      EN2K=EN2B*ENMI(N2K,L2K,M2K,OE2K)                                  TINT
      H2=F2*H(JC,JD,IDAR(3,7))                                          TINT
      X6A=INTEG3(JA,JB,JC,JD)*F3                                        TINT
      DO 1 JE=1,LIM3B                                                   TINT
      N3B=QN(IB3+JE)                                                    TINT
      OE3B=ORBEXP(IB3+JE)                                               TINT
      EN3B=EN2K*ENMI(N3B,L3B,M3B,OE3B)                                  TINT
      DO 1 JF=1,LIM3K                                                   TINT
      N3K=QN(IK3+JF)                                                    TINT
      OE3K=ORBEXP(IK3+JF)                                               TINT
      EN3K=EN3B*ENMI(N3K,L3K,M3K,OE3K)                                  TINT
C     THE INTEGRALS ARE ARRANGED AS:                                    TINT
C  X1=(1/R21)*(1/R13)     X2=(1/R12)*(1/R23)       X3=(1/R31)*(1/R12)   TINT
C  X4=(1/R13)*(1/R32)     X5=(1/R32)*(1/R21)       X6=(1/R23)*(1/R31)   TINT
      X1=REPI(3,N2B,L2B,M2B,OE2B,N3B,L3B,M3B,OE3B,N2K,L2K,M2K,OE2K,N3K,LTINT
     .3K,M3K,OE3K,N1B,L1B,M1B,OE1B,N1K,L1K,M1K,OE1K)                    TINT
      X2=REPI(3,N1B,L1B,M1B,OE1B,N3B,L3B,M3B,OE3B,N1K,L1K,M1K,OE1K,N3K,LTINT
     .3K,M3K,OE3K,N2B,L2B,M2B,OE2B,N2K,L2K,M2K,OE2K)                    TINT
      X3=REPI(3,N3B,L3B,M3B,OE3B,N2B,L2B,M2B,OE2B,N3K,L3K,M3K,OE3K,N2K,LTINT
     .2K,M2K,OE2K,N1B,L1B,M1B,OE1B,N1K,L1K,M1K,OE1K)                    TINT
      X4=REPI(3,N1B,L1B,M1B,OE1B,N2B,L2B,M2B,OE2B,N1K,L1K,M1K,OE1K,N2K,LTINT
     .2K,M2K,OE2K,N3B,L3B,M3B,OE3B,N3K,L3K,M3K,OE3K)                    TINT
      X5=REPI(3,N3B,L3B,M3B,OE3B,N1B,L1B,M1B,OE1B,N3K,L3K,M3K,OE3K,N1K,LTINT
     .1K,M1K,OE1K,N2B,L2B,M2B,OE2B,N2K,L2K,M2K,OE2K)                    TINT
      X6=REPI(3,N2B,L2B,M2B,OE2B,N1B,L1B,M1B,OE1B,N2K,L2K,M2K,OE2K,N1K,LTINT
     .1K,M1K,OE1K,N3B,L3B,M3B,OE3B,N3K,L3K,M3K,OE3K)                    TINT
      X7=H1*INTEG1(JC,JD,JE,JF)                                         TINT
      X8=H2*INTEG2(JA,JB,JE,JF)                                         TINT
      X9=X6A*H(JE,JF,IDAR(5,7))                                         TINT
      X=EN3K*(X1+X2+X3+X4+X5+X6)+X7+X8+X9                               TINT
      INTEG(JA,JB,JC,JD,JE,JF)=X                                        TINT
1     IF(X.EQ.0.D0)NULI=NULI+1                                          TINT
      RC=.TRUE.                                                         TINT
      IF(NULI.LT.LIM1B*LIM1K*LIM2B*LIM2K*LIM3B*LIM3K)RC=.FALSE.         TINT
      NULL3(INTLI3)=RC                                                  TINT
      IF(RC)RETURN                                                      TINT
      CALL NOTE(FDUB,INFO)                                              TINT
      WRITE(1'(3000+INTLI3)*1000)INFO(2),INFO(2),INFO(3),INFO(4)        TINT
      WRITE(2)INTEG                                                     TINT
      RETURN                                                            TINT
100   WRITE(6,900)
```

```
      STOP                                                              TINT
900   FORMAT(' WRONG RETURN IN I/O ROUT')                               TINT
      END                                                               TINT
      SUBROUTINE DENS3(LIMDI3,EXPCOE)                                   DNS3
      IMPLICIT REAL*8(A-H,O-Z)                                          DNS3
      COMMON/SPLI1/I1,I2,J1,J2,K1,K2,L1,L2,LIMI,LIMJ,LIMK,LIML,JMI,JMJ,JDNS3
     .MK,JML,IEXP,JEXP,KEXP,LEXP                                        DNS3
      COMMON/DENSI3/DIJ(15,15),DIK(15,15),DJK(15,15),CDIJK(5,5,15),CDIKJDNS3
     .(5,5,15),CDJKI(5,5,15)                                            DNS3
      EQUIVALENCE(LIMV(1),LIMI),(IXV(1),IEXP)                           DNS3
      INTEGER INFO(4),FDU3,LIMV(3),IXV(3),INXV(50)                      DNS3
      INTEGER*2 LEN2/1800/,LEN3/3000/                                   DNS3
      REAL*8 EXPCOE(5,10)                                               DNS3
      LOGICAL WDH/.FALSE./                                              DNS3
      IJN(I,J)=I+(J*(J-1))/2                                            DNS3
      IF(WDH)GOTO1                                                      DNS3
      WDH=.TRUE.                                                        DNS3
      CALL LOGIOU(INFO,'3          ',&100)                              DNS3
      FDU3=INFO(1)                                                      DNS3
1     DO 2 JA=1,2                                                       DNS3
      IS=JA+1                                                           DNS3
      DO 3 JB=IS,3                                                      DNS3
      INDEX=100*IXV(JA)+10*IXV(JB)                                      DNS3
      IF(LIMDI3.EQ.0)GOTO4                                              DNS3
      DO 5 JD=1,LIMDI3                                                  DNS3
      IF(INXV(JD).EQ.INDEX)GOTO3                                        DNS3
5     CONTINUE                                                          DNS3
4     LIMDI3=LIMDI3+1                                                   DNS3
      IF(LIMDI3.GT.50)GOTO102                                           DNS3
      INXV(LIMDI3)=INDEX                                                DNS3
      LIM1=LIMV(JA)                                                     DNS3
      LIM2=LIMV(JB)                                                     DNS3
      IX1=IXV(JA)                                                       DNS3
      IX2=IXV(JB)                                                       DNS3
      DO 6 JB1=1,LIM1                                                   DNS3
      DO 6 JK1=1,JB1                                                    DNS3
      JBK1=IJN(JK1,JB1)                                                 DNS3
      EXP1=EXPCOE(JB1,IX1)*EXPCOE(JK1,IX1)                              DNS3
      DO 6 JB2=1,LIM2                                                   DNS3
      EXP2=EXP1*EXPCOE(JB2,IX2)                                         DNS3
      DO 6 JK2=1,JB2                                                    DNS3
      JBK2=IJN(JK2,JB2)                                                 DNS3
6     DIJ(JBK1,JBK2)=EXP2*EXPCOE(JK2,IX2)                               DNS3
      CALL NOTE(FDU3,INFO)                                              DNS3
      WRITE(1'INDEX*1000)INFO(2),INFO(2),INFO(3),INFO(4)                DNS3
      CALL WRITE(DIJ,LEN2,0,LENR,3,&101)                                DNS3
3     CONTINUE                                                          DNS3
2     CONTINUE                                                          DNS3
C THE 2-VECTOR DENSITY MATRICES ARE COMPUTED                            DNS3
      DO 10 JA=1,2                                                      DNS3
      IS=JA+1                                                           DNS3
      DO 11 JB=IS,3                                                     DNS3
      JC=JB-JA                                                          DNS3
      IF(JA.EQ.1.AND.JC.EQ.1)JC=3                                       DNS3
      IX1=IXV(JA)                                                       DNS3
      IX2=IXV(JB)                                                       DNS3
      IX3=IXV(JC)                                                       DNS3
      INDEX=IX1*100+IX2*10+IX3                                          DNS3
      DO 12 JD=1,LIMDI3                                                 DNS3
      IF(INDEX.EQ.INXV(JD))GOTO11                                       DNS3
```

```
12    CONTINUE                                                        DNS3
      LIMDI3=LIMDI3+1                                                 DNS3
      IF(LIMDI3.GT.50)GOTO102                                         DNS3
      INXV(LIMDI3)=INDEX                                              DNS3
      LIM1=LIMV(JA)                                                   DNS3
      LIM2=LIMV(JB)                                                   DNS3
      LIM3=LIMV(JC)                                                   DNS3
      DO 13 JD=1,LIM1                                                 DNS3
      DO 13 JE=1,LIM2                                                 DNS3
      EXP1=EXPCOE(JD,IX1)*EXPCOE(JE,IX2)                              DNS3
      DO 13 JF=1,LIM3                                                 DNS3
      EXP2=EXP1*EXPCOE(JF,IX3)                                        DNS3
      DO 13 JG=1,JF                                                   DNS3
      JFG=IJN(JG,JF)                                                  DNS3
13    CDIJK(JD,JE,JFG)=EXP2*EXPCOE(JG,IX3)                            DNS3
      CALL NOTE(FDU3,INFO)                                            DNS3
      WRITE(1'INDEX*1000)INFO(2),INFO(2),INFO(3),INFO(4)              DNS3
      CALL WRITE(CDIJK,LEN3,0,LENR,3,&101)                            DNS3
11    CONTINUE                                                        DNS3
10    CONTINUE                                                        DNS3
C   THE DENSITY MATRICES TO BE USED ARE READ IN                       DNS3
      DO 14 JA=1,2                                                    DNS3
      IS=JA+1                                                         DNS3
      DO 15 JB=IS,3                                                   DNS3
      JC=JB-JA                                                        DNS3
      IF(JA.EQ.1.AND.JC.EQ.1)JC=3                                     DNS3
      INDEX=IXV(JA)*100+IXV(JB)*10                                    DNS3
      READ(1'INDEX*1000)INFO                                          DNS3
      CALL POINT(FDU3,INFO,1)                                         DNS3
      IF(JC-2)18,17,16                                                DNS3
16    CALL READ(DIJ,LEN2,0,LENR,3,&101)                               DNS3
      GOTO19                                                          DNS3
17    CALL READ(DIK,LEN2,0,LENR,3,&101)                               DNS3
      GOTO19                                                          DNS3
18    CALL READ(DJK,LEN2,0,LENR,3,&101)                               DNS3
19    INDEX=IXV(JA)*100+IXV(JB)*10+IXV(JC)                            DNS3
      READ(1'INDEX*1000)INFO                                          DNS3
      CALL POINT(FDU3,INFO,1)                                         DNS3
      IF(JC-2)22,21,20                                                DNS3
20    CALL READ(CDIJK,LEN3,0,LENR,3,&101)                             DNS3
      GOTO15                                                          DNS3
21    CALL READ(CDIKJ,LEN3,0,LENR,3,&101)                             DNS3
      GOTO15                                                          DNS3
22    CALL READ(CDJKI,LEN3,0,LENR,3,&101)                             DNS3
15    CONTINUE                                                        DNS3
14    CONTINUE                                                        DNS3
      RETURN                                                          DNS3
100   WRITE(6,900)                                                    DNS3
      STOP                                                            DNS3
101   WRITE(6,901)                                                    DNS3
      STOP                                                            DNS3
102   WRITE(6,902)                                                    DNS3
      STOP                                                            DNS3
900   FORMAT('  WRONG RETURN FROM LOGIOU')                            DNS3
901   FORMAT('  WRONG RETURN FROM I/O-ROUTINES IN DENS3')             DNS3
902   FORMAT(' DIMENSION OF INXV IN DENS3 EXCEEDED')                  DNS3
      END                                                             DNS3
      SUBROUTINE FOUREL(NORB,NOB,LIMDIX)                              FOUR
      IMPLICIT REAL*8(A-H,O-Z)                                        FOUR
      COMMON/FOUR/FAC4(300),FHH4(5,5,4),LHH4(5,5,4,4),INT4(8,300),LIM4,NFOUR
```

```
.ULL4(100)                                                          FOUR
 COMMON/SPLI1/I1,I2,J1,J2,K1,K2,L1,L2,LIMI,LIMJ,LIMK,LIML,JMI,JMJ,JFOUR
.MK,JML,IEXP,JEXP,KEXP,LEXP                                         FOUR
 COMMON/ALL/EXPCOE(5,10),ORBEXP(15),H(5,5,3),CHARGE,QN,NOBT(3),ISYMFOUR
.,FDUB                                                              FOUR
 COMMON/SYM/IDAR(8,10)                                              FOUR
 COMMON/DENSIT/DIJK(15,15,15),DIJL(15,15,15),DIKL(15,15,15),DJKL(15FOUR
.,15,15),DIJ(15,15),DIK(15,15),DIL(15,15),DJK(15,15),DJL(15,15),DKLFOUR
.(15,15),CIJ(5,5),CIK(5,5),CIL(5,5),CJK(5,5),CJL(5,5),CKL(5,5)     FOUR
 LOGICAL NULL4,LOI,LOJ,LOK,LOL,WDH/.FALSE./,LC(6)                   FOUR
 COMMON/INTRA4/D12(5,5,5,5),D13(5,5,5,5),D14(5,5,5,5),D23(5,5,5,5),FOUR
.D24(5,5,5,5),D34(5,5,5,5)                                          FOUR
 INTEGER QN(15),FDUB,IV(8)                                          FOUR
 REAL*8 LHH4                                                        FOUR
 IJN(I,J)=MINO(I,J)+(MAXO(I,J)*(MAXO(I,J)-1))/2                     FOUR
 IF(WDH)GOTO11                                                      FOUR
 WDH=.TRUE.                                                         FOUR
 DO 12 JA=1,100                                                     FOUR
12 NULL4(JA)=.FALSE.                                                FOUR
11 DO 1 JA=1,LIM4                                                   FOUR
 IF(NULL4(JA))GOTO1                                                 FOUR
 I1=INT4(1,JA)                                                      FOUR
 I2=INT4(2,JA)                                                      FOUR
 J1=INT4(3,JA)                                                      FOUR
 J2=INT4(4,JA)                                                      FOUR
 K1=INT4(5,JA)                                                      FOUR
 K2=INT4(6,JA)                                                      FOUR
 L1=INT4(7,JA)                                                      FOUR
 L2=INT4(8,JA)                                                      FOUR
 CALL SYM34(NOBT,LOI,LOJ,LOK,LOL,ISYM,&1,4)                         FOUR
 CALL FOINT(FDUB,&1,NULL4(JA))                                      FOUR
 ASSIGN 8 TO ICASE                                                  FOUR
 IF(LOK)ASSIGN 7 TO ICASE                                           FOUR
 IF(LOJ)ASSIGN 6 TO ICASE                                           FOUR
 IF(LOI)ASSIGN 5 TO ICASE                                           FOUR
 CALL DENS(LIMDIX,EXPCOE)                                           FOUR
 FACTOR=FAC4(JA)+FAC4(JA)                                           FOUR
 DO 4 IB=1,LIMI                                                     FOUR
 IV(1)=IB                                                           FOUR
 DO 4 IK=1,LIMI                                                     FOUR
 IV(2)=IK                                                           FOUR
 IBK=IJN(IK,IB)                                                     FOUR
 DO 4 JB=1,LIMJ                                                     FOUR
 IV(3)=JB                                                           FOUR
 DO 4 JK=1,LIMJ                                                     FOUR
 IV(4)=JK                                                           FOUR
 JBK=IJN(JK,JB)                                                     FOUR
 DO 4 KB=1,LIMK                                                     FOUR
 IV(5)=KB                                                           FOUR
 DO 4 KK=1,LIMK                                                     FOUR
 IV(6)=KK                                                           FOUR
 KBK=IJN(KK,KB)                                                     FOUR
 DO 4 LB=1,LIML                                                     FOUR
 IV(7)=LB                                                           FOUR
 DO 4 LK=1,LIML                                                     FOUR
 IV(8)=LK                                                           FOUR
 LBK=IJN(LB,LK)                                                     FOUR
 X1=(D12(IV(IDAR(1,1)),IV(IDAR(2,1)),IV(IDAR(3,1)),IV(IDAR(4,1)))*DFOUR
.34(IV(IDAR(5,1)),IV(IDAR(6,1)),IV(IDAR(7,1)),IV(IDAR(8,1)))+D13(IVFOUR
.(IDAR(1,1)),IV(IDAR(2,1)),IV(IDAR(5,1)),IV(IDAR(6,1)))*D24(IV(IDARFOUR
.(IDAR(1,1)),IV(IDAR(2,1)),IV(IDAR(5,1)),IV(IDAR(6,1)))*D24(IV(IDARFOUR
```

```
     .(3,1)),IV(IDAR(4,1)),IV(IDAR(7,1)),IV(IDAR(8,1)))+D14(IV(IDAR(1,1)FOUR
     .),IV(IDAR(2,1)),IV(IDAR(7,1)),IV(IDAR(8,1)))*D23(IV(IDAR(3,1)),IV(FOUR
     .IDAR(4,1)),IV(IDAR(5,1)),IV(IDAR(6,1))))*FACTOR                   FOUR
      GOTO ICASE,(5,6,7,8)                                               FOUR
5     FHH4(IB,IK,JMI)=FHH4(IB,IK,JMI)+X1*DJKL(JBK,KBK,LBK)               FOUR
      X2=X1*DKL(KBK,LBK)                                                 FOUR
      LHH4(IK,JB,JMI,JMJ)=LHH4(IK,JB,JMI,JMJ)+CIJ(IB,JK)*X2              FOUR
      LHH4(IK,JK,JMI,JMJ)=LHH4(IK,JK,JMI,JMJ)+CIJ(IB,JB)*X2              FOUR
      LHH4(IB,JK,JMI,JMJ)=LHH4(IB,JK,JMI,JMJ)+CIJ(IK,JB)*X2              FOUR
      LHH4(IB,JB,JMI,JMJ)=LHH4(IB,JB,JMI,JMJ)+CIJ(IK,JK)*X2              FOUR
      X2=X1*DJL(JBK,LBK)                                                 FOUR
      LHH4(IK,KB,JMI,JMK)=LHH4(IK,KB,JMI,JMK)+CIK(IB,KK)*X2              FOUR
      LHH4(IK,KK,JMI,JMK)=LHH4(IK,KK,JMI,JMK)+CIK(IB,KB)*X2              FOUR
      LHH4(IB,KK,JMI,JMK)=LHH4(IB,KK,JMI,JMK)+CIK(IK,KB)*X2              FOUR
      LHH4(IB,KB,JMI,JMK)=LHH4(IB,KB,JMI,JMK)+CIK(IK,KK)*X2              FOUR
      X2=X1*DJK(JBK,KBK)                                                 FOUR
      LHH4(IK,LB,JMI,JML)=LHH4(IK,LB,JMI,JML)+CIL(IB,LK)*X2              FOUR
      LHH4(IK,LK,JMI,JML)=LHH4(IK,LK,JMI,JML)+CIL(IB,LB)*X2              FOUR
      LHH4(IB,LK,JMI,JML)=LHH4(IB,LK,JMI,JML)+CIL(IK,LB)*X2              FOUR
      LHH4(IB,LB,JMI,JML)=LHH4(IB,LB,JMI,JML)+CIL(IK,LK)*X2              FOUR
6     FHH4(JB,JK,JMJ)=FHH4(JB,JK,JMJ)+X1*DIKL(IBK,KBK,LBK)               FOUR
      X2=X1*DIL(IBK,LBK)                                                 FOUR
      LHH4(JK,KB,JMJ,JMK)=LHH4(JK,KB,JMJ,JMK)+CJK(JB,KK)*X2              FOUR
      LHH4(JK,KK,JMJ,JMK)=LHH4(JK,KK,JMJ,JMK)+CJK(JB,KB)*X2              FOUR
      LHH4(JB,KK,JMJ,JMK)=LHH4(JB,KK,JMJ,JMK)+CJK(JK,KB)*X2              FOUR
      LHH4(JB,KB,JMJ,JMK)=LHH4(JB,KB,JMJ,JMK)+CJK(JK,KK)*X2              FOUR
      X2=X1*DIK(IBK,KBK)                                                 FOUR
      LHH4(JK,LB,JMJ,JML)=LHH4(JK,LB,JMJ,JML)+CJL(JB,LK)*X2              FOUR
      LHH4(JK,LK,JMJ,JML)=LHH4(JK,LK,JMJ,JML)+CJL(JB,LB)*X2              FOUR
      LHH4(JB,LK,JMJ,JML)=LHH4(JB,LK,JMJ,JML)+CJL(JK,LB)*X2              FOUR
      LHH4(JB,LB,JMJ,JML)=LHH4(JB,LB,JMJ,JML)+CJL(JK,LK)*X2              FOUR
7     FHH4(KB,KK,JMK)=FHH4(KB,KK,JMK)+X1*DIJL(IBK,JBK,LBK)               FOUR
      X2=X1*DIJ(IBK,JBK)                                                 FOUR
      LHH4(KK,LB,JMK,JML)=LHH4(KK,LB,JMK,JML)+CKL(KB,LK)*X2              FOUR
      LHH4(KK,LK,JMK,JML)=LHH4(KK,LK,JMK,JML)+CKL(KB,LB)*X2              FOUR
      LHH4(KB,LK,JMK,JML)=LHH4(KB,LK,JMK,JML)+CKL(KK,LB)*X2              FOUR
      LHH4(KB,LB,JMK,JML)=LHH4(KB,LB,JMK,JML)+CKL(KK,LK)*X2              FOUR
8     FHH4(LB,LK,JML)=FHH4(LB,LK,JML)+X1*DIJK(IBK,JBK,KBK)               FOUR
4     CONTINUE                                                           FOUR
1     CONTINUE                                                           FOUR
      DO 21 JA=1,NORB                                                    FOUR
      DO 22 JB=1,NOB                                                     FOUR
      DO 22 JC=1,JB                                                      FOUR
      FHH4(JB,JC,JA)=0.25D0*(FHH4(JB,JC,JA)+FHH4(JC,JB,JA))              FOUR
      FHH4(JC,JB,JA)=FHH4(JB,JC,JA)                                      FOUR
      LHH4(JC,JB,JA,JA)=(LHH4(JC,JB,JA,JA)+LHH4(JB,JC,JA,JA))*0.125D0    FOUR
22    LHH4(JB,JC,JA,JA)=LHH4(JC,JB,JA,JA)                                FOUR
      IS=JA+1                                                            FOUR
      IF(IS.GT.NORB)RETURN                                               FOUR
      DO 21 JB=IS,NORB                                                   FOUR
      DO 21 JC=1,NOB                                                     FOUR
      DO 21 JD=1,NOB                                                     FOUR
      LHH4(JC,JD,JA,JB)=LHH4(JC,JD,JA,JB)*0.125D0                        FOUR
21    LHH4(JD,JC,JB,JA)=LHH4(JC,JD,JA,JB)                                FOUR
      RETURN                                                             FOUR
      END                                                                FOUR
      SUBROUTINE FOINT(FDUB,*,NULL)                                      FOIN
      IMPLICIT REAL*8(A-H,O-Z)                                           FOIN
      REAL*8 LH2,LHH2                                                    FOIN
      COMMON/TWO/FH2(5,5,4),FHH2(5,5,4),FAC2(100),LH2(5,5,4,4),LHH2(5,   FOIN
```

```
     .5,4,4),INT2(4,100),INTNO2(100),NULL2(100),LIM2,INTLI2               FOIN
      COMMON/SYM/IDAR(8,10)                                                FOIN
      LOGICAL NULL,RC,LC(6),NULL2                                          FOIN
      INTEGER IT(4,6)/1,2,3,4,1,2,5,6,1,2,7,8,3,4,5,6,3,4,7,8,5,6,7,8/,QFOIN
     .N(15),FDUB,IV(8)                                                     FOIN
10    DO 2 JB=1,6                                                          FOIN
      DO 3 JC=1,INTLI2                                                     FOIN
      IF(IDAR(JB,8).NE.INTNO2(JC))GOTO3                                    FOIN
      LC(JB)=NULL2(JC)                                                     FOIN
      GOTO31                                                               FOIN
3     CONTINUE                                                             FOIN
      INTLI2=INTLI2+1                                                      FOIN
      INTNO2(INTLI2)=IDAR(JB,8)                                            FOIN
      CALL TWINT(4,IT(1,JB),IT(2,JB),IT(3,JB),IT(4,JB),RC)                 FOIN
      LC(JB)=RC                                                            FOIN
      JC=INTLI2                                                            FOIN
31    CALL LIES(LC,JC,JB,FDUB)                                             FOIN
2     CONTINUE                                                             FOIN
      NULL=(LC(1).OR.LC(6)).AND.(LC(2).OR.LC(5)).AND.(LC(3).OR.LC(4))      FOIN
      IF(NULL)RETURN1                                                      FOIN
      RETURN                                                               FOIN
      END                                                                  FOIN
      SUBROUTINE DENS (LIMDIX,EXPCOE)                                      DNS4
      IMPLICIT REAL*8(A-H,O-Z)                                             DNS4
      COMMON/SPLI1/I1,I2,J1,J2,K1,K2,L1,L2,LIMI,LIMJ,LIMK,LIML,JMI,JMJ,JDNS4
     .MK,JML,IEXP,JEXP,KEXP,LEXP                                           DNS4
      COMMON/DENSIT/DIJK(15,15,15),DIJL(15,15,15),DIKL(15,15,15),DJKL(15DNS4
     .,15,15),DIJ(15,15),DIK(15,15),DIL(15,15),DJK(15,15),DJL(15,15),DKLDNS4
     .(15,15),CIJ(5,5),CIK(5,5),CIL(5,5),CJK(5,5),CJL(5,5),CKL(5,5)       DNS4
      INTEGER LIMV(4),IXV(4),INXV(50),INFO(4),FDU,IM2(3,4)/1,2,3,1,2,4,1DNS4
     .,3,4,2,3,4/                                                          DNS4
      EQUIVALENCE(LIMV(1),LIMI),(IXV(1),IEXP)                              DNS4
      INTEGER*2 LEN1/27000/,LEN2/1800/,LEN3/200/                           DNS4
      LOGICAL WDH/.FALSE./                                                 DNS4
      REAL*8 EXPCOE(5,10)                                                  DNS4
      IJN(I,J)=I+(J*(J-1))/2                                               DNS4
      IF(WDH)GOTO1                                                         DNS4
      WDH=.TRUE.                                                           DNS4
      CALL LOGIOU(INFO,'3        ',&101)                                   DNS4
      FDU=INFO(1)                                                          DNS4
C   THE SIX-VECTOR DENSITY MATRICES ARE COMPUTED AND WRITTEN ON UNIT(3)   DNS4
1     DO 2 JA=1,2                                                          DNS4
      IS1=JA+1                                                             DNS4
      DO 3 JB=IS1,3                                                        DNS4
      IS2=JB+1                                                             DNS4
      DO 4 JC=IS2,4                                                        DNS4
      INDEX=100*IXV(JA)+10*IXV(JB)+IXV(JC)                                 DNS4
      IF(LIMDIX.EQ.0)GOTO5                                                 DNS4
      DO 6 JD=1,LIMDIX                                                     DNS4
      IF(INXV(JD).EQ.INDEX)GOTO4                                           DNS4
6     CONTINUE                                                             DNS4
5     LIMDIX=LIMDIX+1                                                      DNS4
      IF(LIMDIX.GT.50)GOTO102                                              DNS4
      INXV(LIMDIX)=INDEX                                                   DNS4
      CALL NOTE(FDU,INFO)                                                  DNS4
      WRITE(1'INDEX*1000)INFO(2),INFO(2),INFO(3),INFO(4)                   DNS4
      LIM1=LIMV(JA)                                                        DNS4
      LIM2=LIMV(JB)                                                        DNS4
      LIM3=LIMV(JC)                                                        DNS4
      IX1=IXV(JA)
```

```
      IX2=IXV(JB)                                                       DNS4
      IX3=IXV(JC)                                                       DNS4
      DO 7 JB1=1,LIM1                                                   DNS4
      DO 7 JK1=1,JB1                                                    DNS4
      JBK1=IJN(JK1,JB1)                                                 DNS4
      EXP1=EXPCOE(JB1,IX1)*EXPCOE(JK1,IX1)                              DNS4
      DO 7 JB2=1,LIM2                                                   DNS4
      EXP2=EXP1*EXPCOE(JB2,IX2)                                         DNS4
      DO 7 JK2=1,JB2                                                    DNS4
      JBK2=IJN(JK2,JB2)                                                 DNS4
      EXP3=EXP2*EXPCOE(JK2,IX2)                                         DNS4
      DO 7 JB3=1,LIM3                                                   DNS4
      EXP4=EXP3*EXPCOE(JB3,IX3)                                         DNS4
      DO 7 JK3=1,JB3                                                    DNS4
      JBK3=IJN(JK3,JB3)                                                 DNS4
      DJKL(JBK1,JBK2,JBK3)=EXPCOE(JK3,IX3)*EXP4                         DNS4
      CALL WRITE(DJKL,LEN1,0,LENR,3,&100)                               DNS4
      CONTINUE                                                          DNS4
      CONTINUE                                                          DNS4
      CONTINUE                                                          DNS4
   THE DENSITY MATRICES CONTAINING FOUR VECTORS ARE COMPUTED AND WRITTEDNS4
      DO 10 JA=1,3                                                      DNS4
      IS=JA+1                                                           DNS4
      DO 11 JB=IS,4                                                     DNS4
      IX1=IXV(JA)                                                       DNS4
      IX2=IXV(JB)                                                       DNS4
      INDEX=(IX1*10+IX2)*10                                             DNS4
      DO 12 JC=1,LIMDIX                                                 DNS4
      IF(INXV(JC).EQ.INDEX)GOTO11                                       DNS4
12    CONTINUE                                                          DNS4
      LIMDIX=LIMDIX+1                                                   DNS4
      INXV(LIMDIX)=INDEX                                                DNS4
      LIM1=LIMV(JA)                                                     DNS4
      LIM2=LIMV(JB)                                                     DNS4
      DO 13 JB1=1,LIM1                                                  DNS4
      DO 13 JK1=1,JB1                                                   DNS4
      JBK1=IJN(JK1,JB1)                                                 DNS4
      EXP1=EXPCOE(JB1,IX1)*EXPCOE(JK1,IX1)                              DNS4
      DO 13 JB2=1,LIM2                                                  DNS4
      EXP2=EXP1*EXPCOE(JB2,IX2)                                         DNS4
      DO 13 JK2=1,JB2                                                   DNS4
      JBK2=IJN(JK2,JB2)                                                 DNS4
13    DIJ(JBK1,JBK2)=EXP2*EXPCOE(JK2,IX2)                               DNS4
      CALL NOTE(FDU,INFO)                                               DNS4
      WRITE(1'INDEX*1000)INFO(2),INFO(2),INFO(3),INFO(4)                DNS4
      CALL WRITE(DIJ,LEN2,0,LENR,3,&100)                                DNS4
11    CONTINUE                                                          DNS4
10    CONTINUE                                                          DNS4
  THE MIXED DENSITY-MATRICES OF TWO VECTORS ARE COMPUTED                DNS4
      IREP=0                                                            DNS4
      DO 20 JA=1,3                                                      DNS4
      IS=JA+1                                                           DNS4
      DO 21 JB=IS,4                                                     DNS4
      IREP=IREP+1                                                       DNS4
      IX1=IXV(JA)                                                       DNS4
      IX2=IXV(JB)                                                       DNS4
      INDEX=IX1*10+IX2                                                  DNS4
      DO 22 JC=1,LIMDIX                                                 DNS4
      IF(INXV(JC).EQ.INDEX)GOTO21                                       DNS4
22    CONTINUE
```

```
      LIMDIX=LIMDIX+1                                                   DNS4
      INXV(LIMDIX)=INDEX                                                DNS4
      LIM1=LIMV(JA)                                                     DNS4
      LIM2=LIMV(JB)                                                     DNS4
      DO 23 J1B=1,LIM1                                                  DNS4
      DO 23 J1K=1,LIM2                                                  DNS4
23    CIJ(J1B,J1K)=EXPCOE(J1B,IX1)*EXPCOE(J1K,IX2)                      DNS4
      CALL NOTE(FDU,INFO)                                               DNS4
      WRITE(1'INDEX*1000)INFO(2),INFO(2),INFO(3),INFO(4)                DNS4
      CALL WRITE(CIJ,LEN3,0,LENR,3,&100)                                DNS4
21    CONTINUE                                                          DNS4
20    CONTINUE                                                          DNS4
C   THE DENSITY MATRICES TO BE USED IN FOUREL ARE READ IN               DNS4
      DO 30 JA=1,4                                                      DNS4
      INDEX=IXV(IM2(1,JA))*100+IXV(IM2(2,JA))*10+IXV(IM2(3,JA))         DNS4
      READ(1'INDEX*1000)INFO                                            DNS4
      CALL POINT(FDU,INFO,1)                                            DNS4
      GOTO(31,32,33,34),JA                                              DNS4
31    CALL READ(DIJK,LEN1,0,LENR,3,&100)                                DNS4
      GOTO30                                                            DNS4
32    CALL READ(DIJL,LEN1,0,LENR,3,&100)                                DNS4
      GOTO30                                                            DNS4
33    CALL READ(DIKL,LEN1,0,LENR,3,&100)                                DNS4
      GOTO30                                                            DNS4
34    CALL READ(DJKL,LEN1,0,LNR,&100)                                   DNS4
30    CONTINUE                                                          DNS4
C   THE DENSITY MATRICES WITH TWO SUBSCRIPTS ARE READ IN                DNS4
      IREP=0                                                            DNS4
      DO 40 JA=1,3                                                      DNS4
      IS=JA+1                                                           DNS4
      DO 40 JB=IS,4                                                     DNS4
      INDEX=IXV(JA)*10+IXV(JB)                                          DNS4
      IREP=IREP+1                                                       DNS4
      READ(1'INDEX*1000)INFO                                            DNS4
      CALL POINT(FDU,INFO,1)                                            DNS4
      GOTO(41,42,43,44,45,46),IREP                                      DNS4
41    CALL READ(CIJ,LEN3,0,LENR,3,&100)                                 DNS4
      GOTO47                                                            DNS4
42    CALL READ(CIK,LEN3,0,LENR,3,&100)                                 DNS4
      GOTO47                                                            DNS4
43    CALL READ(CIL,LEN3,0,LENR,3,&100)                                 DNS4
      GOTO47                                                            DNS4
44    CALL READ(CJK,LEN3,0,LENR,3,&100)                                 DNS4
      GOTO47                                                            DNS4
45    CALL READ(CJL,LEN3,0,LENR,3,&100)                                 DNS4
      GOTO47                                                            DNS4
46    CALL READ(CKL,LEN3,0,LENR,3,&100)                                 DNS4
47    INDEX=INDEX*10                                                    DNS4
      READ(1'INDEX*1000)INFO                                            DNS4
      CALL POINT(FDU,INFO,1)                                            DNS4
      GOTO(51,52,53,54,55,56),IREP                                      DNS4
51    CALL READ(DIJ,LEN2,0,LENR,3,&100)                                 DNS4
      GOTO40                                                            DNS4
52    CALL READ(DIK,LEN2,0,LENR,3,&100)                                 DNS4
      GOTO40                                                            DNS4
53    CALL READ(DIL,LEN2,0,LENR,3,&100)                                 DNS4
      GOTO40                                                            DNS4
54    CALL READ(DJK,LEN2,0,LENR,3,&100)                                 DNS4
      GOTO40                                                            DNS4
55    CALL READ(DJL,LEN2,0,LENR,3,&100)
```

```
      GOTO40
56    CALL READ(DKL,LEN2,0,LENR,3,&100)                                DNS4
40    CONTINUE                                                         DNS4
      RETURN                                                           DNS4
100   WRITE(6,900)                                                     DNS4
      STOP                                                             DNS4
101   WRITE(6,901)                                                     DNS4
      STOP                                                             DNS4
102   WRITE(6,902)                                                     DNS4
      STOP                                                             DNS4
900   FORMAT(' WRONG RETURN IN I/O-ROUT IN DENS')                      DNS4
901   FORMAT(' LOGIOU HAS WRONG RETURN IN DENS')                       DNS4
902   FORMAT(' DIMENSION OF INXV IN DENS EXCEEDED')                    DNS4
      END                                                              DNS4
      SUBROUTINE LIES(LC,JCI,JBI,FDUB)                                 LIES
      IMPLICIT REAL*8(A-H,O-Z)                                         LIES
      INTEGER*2 LEN/5000/                                              LIES
      LOGICAL LC(6)                                                    LIES
      COMMON/INTRA4/D12(5,5,5,5),D13(5,5,5,5),D14(5,5,5,5),D23(5,5,5,5),LIES
     .D24(5,5,5,5),D34(5,5,5,5)                                        LIES
      INTEGER FDUB,INFO(4),IREP/0/                                     LIES
      IF(IREP.EQ.1)GOTO20                                              LIES
      IREP=1                                                           LIES
      DO 21 JA=1,5                                                     LIES
      DO 21 JB=1,5                                                     LIES
      DO 21 JC=1,5                                                     LIES
      DO 21 JD=1,5                                                     LIES
21    D12(JA,JB,JC,JD)=0.D0                                            LIES
      WRITE(1'10000000)D12                                             LIES
20    IF(.NOT.LC(JBI))GOTO1                                            LIES
      GOTO(2,3,4,5,6,7),JBI                                            LIES
2     READ(1'10000000)D12                                              LIES
      RETURN                                                           LIES
3     READ(1'10000000)D13                                              LIES
      RETURN                                                           LIES
4     READ(1'10000000)D14                                              LIES
      RETURN                                                           LIES
5     READ(1'10000000)D23                                              LIES
      RETURN                                                           LIES
6     READ(1'10000000)D24                                              LIES
      RETURN                                                           LIES
7     READ(1'10000000)D34                                              LIES
      RETURN                                                           LIES
1     READ(1'(2000+JCI)*1000)INFO                                      LIES
      CALL POINT(FDUB,INFO,1)                                          LIES
      GOTO(8,9,10,11,12,13),JBI                                        LIES
8     CALL READ(D12,LEN,0,LNR,2,&100)                                  LIES
      RETURN                                                           LIES
9     CALL READ(D13,LEN,0,LNR,2,&100)                                  LIES
      RETURN                                                           LIES
10    CALL READ(D14,LEN,0,LNR,2,&100)                                  LIES
      RETURN                                                           LIES
11    CALL READ(D23,LEN,0,LNR,2,&100)                                  LIES
      RETURN                                                           LIES
12    CALL READ(D24,LEN,0,LNR,2,&100)                                  LIES
      RETURN                                                           LIES
13    CALL READ(D34,LEN,0,LNR,2,&100)                                  LIES
      RETURN                                                           LIES
100   WRITE(6,900)                                                     LIES
      STOP
```

```
300   FORMAT( 'WRONG RETURN IN I/O ROUT IN LIES,FOUREL')                 LIES
      END                                                                LIES
      SUBROUTINE COMBIN(METHOD,ISYM,ORB,NOBT,FH1,FHH1,FH2,FHH2,FHH3,FHH4COMB
     .,LH2,LHH2,LHH3,LHH4,WK,EXPCOE,EXH,EXHH,TAU)                        COMB
      IMPLICIT REAL*8(A-H,O-Z)                                           COMB
C THIS ROUTINE SETS UP THE F&L-MATRICES AS REQUIRED BY HINZE             COMB
      COMMON/HINZ/S,F,L,NOB,NORB,CLOSED                                  COMB
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                            COMB
C   METHOD IS A PARAMETER,READ BY THE MAIN LINE,THAT DETERMINES THE QUANCOMB
C   BE MINIMIZED                                                         COMB
C  METHOD=1                                                              COMB
C   MINIMIZE:    <H>                       CONSTRAINT:   1               COMB
C  METHOD=2                                                              COMB
C   MINIMIZE:    <(H-E)**2>                CONSTRAINT:   1               COMB
C  METHOD=3                                                              COMB
C   MINIMIZE:    <(H-WK)**2>               CONSTRAINT:   1               COMB
C  METHOD=4                                                              COMB
C   MINIMIZE:    <H-WK>**2/<(H-WK)**2>     CONSTRAINT:   1               COMB
C  METHOD=5                                                              COMB
C   MINIMIZE:    <H-WK>**2/<(H-E)**2>      CONSTRAINT:   1               COMB
      REAL*8 FH1(5,5,4),FHH1(5,5,4),FH2(5,5,4),FHH2(5,5,4),FHH3(5,5,4),FCOMB
     .HH4(5,5,4),LH2(5,5,4,4),LHH2(5,5,4,4),LHH3(5,5,4,4),LHH4(5,5,4,4),COMB
     .F(5,5,4),L(5,5,4,4),EXPCOE(5,10),S(5,5,3),EXH(3,4),EXHH(3,4)       COMB
      LOGICAL CLOSED(3,4)                                                COMB
      INTEGER ORB(3),NOBT(3)                                             COMB
      NOB=NOBT(ISYM)                                                     COMB
      NORB=ORB(ISYM)                                                     COMB
      IST=ISTA(ISYM)                                                     COMB
      GOTO(1,2,3,4,4,4),METHOD                                           COMB
C   FIRST VARIATIONAL SCHEME                                             COMB
1     DO 5 JA=1,NOB                                                      COMB
      DO 5 JB=1,NOB                                                      COMB
      DO 5 JC=1,NORB                                                     COMB
      F(JA,JB,JC)=FH1(JA,JB,INNOR(IST+JC))+FH2(JA,JB,INNOR(IST+JC))      COMB
      DO 5 JD=1,NORB                                                     COMB
5     L(JA,JB,JC,JD)=0.5D0*LH2(JA,JB,INNOR(IST+JC),INNOR(IST+JD))        COMB
      RETURN                                                             COMB
C   SECOND VARIATIONAL SCHEME                                            COMB
2     CALL ENER(ISYM,EXPCOE,FH1,FH2,NOBT,ORB,EXH)                        COMB
      EVH=0.D0                                                           COMB
      DO 8 JA=1,3                                                        COMB
      LINORB=ORB(JA)                                                     COMB
      IF(LINORB.EQ.0)GOTO8                                               COMB
      DO 9 JB=1,LINORB                                                   COMB
9     EVH=EVH+EXH(JA,JB)                                                 COMB
8     CONTINUE                                                           COMB
      GOTO31                                                             COMB
C   THIRD VARIATIONAL SCHEME                                             COMB
3     EVH=WK                                                             COMB
31    FACT=2.D0*EVH                                                      COMB
      DO 6 JA=1,NOB                                                      COMB
      DO 6 JB=1,NOB                                                      COMB
      DO 6 JC=1,NORB                                                     COMB
      JE=INNOR(IST+JC)                                                   COMB
      F(JA,JB,JC)=FHH1(JA,JB,JE)+FHH2(JA,JB,JE)+FHH3(JA,JB,JE)+FHH4(JA,JCOMB
     .B,JE)-FACT*(FH1(JA,JB,JE)+FH2(JA,JB,JE))                           COMB
      DO 6 JD=1,NORB                                                     COMB
      JF=INNOR(IST+JD)                                                   COMB
6     L(JA,JB,JC,JD)=(LHH2(JA,JB,JE,JF)+LHH3(JA,JB,JE,JF)+LHH4(JA,JB,JE,COMB
     .JF)-FACT*LH2(JA,JB,JE,JF))*0.5D0                                   COMB
```

```
      RETURN
C   FOURTH+FIFTH+SIXTH VARIATIONAL SCHEME                                COMB
4     CALL ENER(ISYM,EXPCOE,FH1,FH2,NOBT,ORB,EXH)                        COMB
      CALL EXVAHH(FHH1,FHH2,FHH3,FHH4,EXPCOE,ISYM,NOBT,ORB,EXHH)         COMB
      EVH=0.D0                                                           COMB
      EVHH=0.D0                                                          COMB
      DO 10 JA=1,3                                                       COMB
      LINORB=ORB(JA)                                                     COMB
      IF(LINORB.EQ.0)GOTO10                                              COMB
      DO 11 JB=1,LINORB                                                  COMB
      EVH=EVH+EXH(JA,JB)                                                 COMB
11    EVHH=EVHH+EXHH(JA,JB)                                              COMB
10    CONTINUE                                                           COMB
      DELTA=EVHH-EVH*EVH                                                 COMB
      DELTAS=EVHH-2.D0*EVH*WK+WK*WK                                      COMB
      EPSILO=DABS(EVH-WK)                                                COMB
      EPDS=EPSILO/DELTAS                                                 COMB
      IF(METHOD.EQ.5)EPDS=EPSILO/DELTA                                   COMB
      OMEGA1=2.D0*(TAU*EPSILO-WK)                                        COMB
      OMEGA2=1.D0                                                        COMB
      IF(METHOD.EQ.6)GOTO12                                              COMB
      OMEGA1=2.D0*EPDS*(1.D0+WK*EPDS)                                    COMB
      OMEGA2=-EPDS*EPDS                                                  COMB
12    CONTINUE                                                           COMB
      DO 7 JA=1,NOB                                                      COMB
      DO 7 JB=1,NOB                                                      COMB
      DO 7 JC=1,NORB                                                     COMB
      JE=INNOR(IST+JC)                                                   COMB
      F(JA,JB,JC)=OMEGA2*(FHH1(JA,JB,JE)+FHH2(JA,JB,JE)+FHH3(JA,JB,JE)+FCOMB
     .HH4(JA,JB,JE))+OMEGA1*(FH1(JA,JB,JE)+FH2(JA,JB,JE))                COMB
      DO 7 JD=1,NORB                                                     COMB
      JF=INNOR(IST+JD)                                                   COMB
7     L(JA,JB,JC,JD)=(OMEGA2*(LHH2(JA,JB,JE,JF)+LHH3(JA,JB,JE,JF)+LHH4(JCOMB
     .A,JB,JE,JF))+OMEGA1*LH2(JA,JB,JE,JF))*5.D-1                        COMB
      RETURN                                                             COMB
      DEBUG UNIT(9)                                                      COMB
      AT12                                                               COMB
      DISPLAY EVH,EVHH,DELTA,EPSILO,OMEGA1,OMEGA2                        COMB
      END                                                                COMB
      SUBROUTINE DIAGO(EXPCOE,NOBT,ISYM,FH1,FH2,FHH1,FHH2,FHH3,FHH4,WK) DIAG
      IMPLICIT REAL*8(A-H,O-Z)                                           DIAG
      INTEGER NOBT(3)                                                    DIAG
      COMMON/HINZ/S,F,L,NOB,NORB,CLOSED                                  DIAG
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                            DIAG
      REAL*8 EXPCOE(5,10),S(5,5,3),F(5,5,4),L(5,5,4,4),MAT(15),EIGVEC(5,DIAG
     .5),SMS(15),SMO(15),T(15),FH1(5,5,4),FH2(5,5,4),FHH1(5,5,4),FHH2(5,DIAG
     .5,4),FHH3(5,5,4),FHH4(5,5,4),TEXT(2)/'<H>     ','<H**2>   '/       DIAG
      REAL*8 SMO1(15),SMO2(15),EIGM(25),EIGVAL(5),R(5,5),RV(5)           DIAG
      LOGICAL CLOSED(3,4),NOSTA/.FALSE./                                 DIAG
9999  IF(NOSTA)GOTO10                                                    DIAG
      NOSTA=.TRUE.                                                       DIAG
      DO 12 JA=1,2                                                       DIAG
      DO 11 JB=1,NOB                                                     DIAG
      DO 11 JC=1,JB                                                      DIAG
      IJN=JC+JB*(JB-1)/2                                                 DIAG
11    SMS(IJN)=S(JB,JC,JA)                                               DIAG
      IF(JA.EQ.1)CALL SOMS(NOBT(JA),SMS,SMO1)                            DIAG
      IF(JA.EQ.2)CALL SOMS(NOBT(JA),SMS,SMO2)                            DIAG
12    CONTINUE                                                           DIAG
10    IN=INNO(ISYM)
```

```
C FILL UP THE MATRIX TO BE DIAGONALIZED                                 DIAG
      DO 1 JA=1,NOB                                                     DIAG
      DO 1 JB=1,JA                                                      DIAG
      IJN=JB+JA*(JA-1)/2                                                DIAG
1     MAT(IJN)=F(JA,JB,1)                                               DIAG
      IF(ISYM.EQ.1)CALL MULTS(NOB,MAT,SMO1,T)                           DIAG
      IF(ISYM.EQ.2)CALL MULTS(NOB,MAT,SMO2,T)                           DIAG
      CALL DEIGE(MAT,EIGM,NOB,0)                                        DIAG
      DO 4 JA=1,NOB                                                     DIAG
      DO 4 JB=1,NOB                                                     DIAG
      IK=(JA-1)*NOB+JB                                                  DIAG
4     EIGVEC(JB,JA)=EIGM(IK)                                            DIAG
      IF(ISYM.EQ.1)CALL VMULT(NOB,EIGVEC,SMO1,5)                        DIAG
      IF(ISYM.EQ.2)CALL VMULT(NOB,EIGVEC,SMO2,5)                        DIAG
      MAT(2)=MAT(3)                                                     DIAG
      MAT(3)=MAT(6)                                                     DIAG
      MAT(4)=MAT(10)                                                    DIAG
      MAT(5)=MAT(15)                                                    DIAG
      WRITE(12,900)(MAT(JA),JA=1,NOB)                                   DIAG
      WRITE(12,901)                                                     DIAG
      DO 2 JA=1,NOB                                                     DIAG
2     WRITE(12,900)(EIGVEC(JA,JB),JB=1,NOB)                             DIAG
      WRITE(12,901)                                                     DIAG
900   FORMAT(5D15.5)                                                    DIAG
901   FORMAT(///)                                                       DIAG
C   ADD THE F-MATRICES                                                  DIAG
      IMULT=1                                                           DIAG
      DO 5 JB=1,NOB                                                     DIAG
      DO 5 JC=1,NOB                                                     DIAG
5     R(JB,JC)=0.5D0*FH2(JB,JC,INNOR(ISYM))+FH1(JB,JC,INNOR(ISYM))      DIAG
13    DO 6 JA=1,NOB                                                     DIAG
      EIGVAL(JA)=0.D0                                                   DIAG
      DO 7 JB=1,NOB                                                     DIAG
      RV(JB)=0.D0                                                       DIAG
      DO 7 JC=1,NOB                                                     DIAG
7     RV(JB)=RV(JB)+EIGVEC(JC,JA)*R(JC,JB)                              DIAG
      DO 8 JB=1,NOB                                                     DIAG
8     EIGVAL(JA)=EIGVAL(JA)+RV(JB)*EIGVEC(JB,JA)                        DIAG
6     CONTINUE                                                          DIAG
      WRITE(12,902)TEXT(IMULT),(EIGVAL(JA),JA=1,NOB)                    DIAG
      IMULT=IMULT+1                                                     DIAG
      IF(IMULT.EQ.3)GOTO14                                              DIAG
      DAMIN=DABS(WK-EIGVAL(1))                                          DIAG
      ISK=1                                                             DIAG
      DO 15 JA=2,NOB                                                    DIAG
      DBMIN=DABS(WK-EIGVAL(JA))                                         DIAG
      IF(DAMIN.LE.DBMIN)GOTO15                                          DIAG
      DAMIN=DBMIN                                                       DIAG
      ISK=JA                                                            DIAG
15    CONTINUE                                                          DIAG
      JC=INNOR(ISYM)                                                    DIAG
      DO 9 JA=1,NOB                                                     DIAG
      DO 9 JB=1,NOB                                                     DIAG
9     R(JA,JB)=FHH1(JA,JB,JC)+0.5D0*FHH2(JA,JB,JC)+FHH3(JA,JB,JC)/3.D0+FDIAG
     .HH4(JA,JB,JC)*0.25D0                                              DIAG
      GOTO13                                                            DIAG
902   FORMAT(1H ,A8,5D20.10)                                            DIAG
904   FORMAT(I1)                                                        DIAG
14    DO 3 JA=1,NOB                                                     DIAG
3     EXPCOE(JA,IN)=EIGVEC(JA,ISK)                                      DIAG
```

```
      RETURN
      DEBUG UNIT(9),SUBTRACE,SUBCHK                                     DIAG
      END                                                               DIAG
      SUBROUTINE CNVRGC(EXPCOE,ITER,NOBT,ORB,*)                         DIAG
      IMPLICIT REAL*8(A-H,O-Z)                                          CNVR
      INTEGER NOBT(3),ORB(3)                                            CNVR
      REAL*8 EXPCOE(5,10),OLDEXP(5,10)                                  CNVR
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                           CNVR
      IORB=ORB(1)+ORB(2)+ORB(3)                                         CNVR
      IF (ITER.EQ.1)GOTO10                                              CNVR
      SUM=0.D0                                                          CNVR
      DO 1 JA=1,IORB                                                    CNVR
C   LIM DOES NOT INCLUDE D-ORBITALS                                     CNVR
      LIM=NOBT(INNO(JA)/5+1)                                            CNVR
      DO 1 JB=1,LIM                                                     CNVR
1     SUM=SUM+(EXPCOE(JB,INNO(JA))-OLDEXP(JB,INNO(JA)))**2              CNVR
      SUM=DSQRT(SUM)                                                    CNVR
      WRITE(11,900)ITER,SUM                                             CNVR
900   FORMAT(' ITERATION ',I3,' CONV.SUM=',D15.5)                       CNVR
      IF(SUM.LT.1.D-8)RETURN1                                           CNVR
      DO 12 JA=1,IORB                                                   CNVR
      LIM=NOBT(INNO(JA)/5+1)                                            CNVR
      SMAX=DABS(EXPCOE(1,INNO(JA)))                                     CNVR
      ISK=1                                                             CNVR
      DO 13 JB=2,LIM                                                    CNVR
      IF(DABS(EXPCOE(JB,INNO(JA))).LE.SMAX)GOTO13                       CNVR
      ISK=JB                                                            CNVR
      SMAX=DABS(EXPCOE(JB,INNO(JA)))                                    CNVR
13    CONTINUE                                                          CNVR
      SSIGN=DSIGN(1.D0,EXPCOE(ISK,INNO(JA)))                            CNVR
      DO 14 JB=1,LIM                                                    CNVR
14    EXPCOE(JB,INNO(JA))=0.5D0*(OLDEXP(JB,INNO(JA))+SSIGN*EXPCOE(JB,INNCNVR
     .O(JA)))                                                           CNVR
12    CONTINUE                                                          CNVR
10    DO 11 JA=1,IORB                                                   CNVR
      LIM=NOBT(INNO(JA)/5+1)                                            CNVR
      DO 11 JB=1,LIM                                                    CNVR
11    OLDEXP(JB,INNO(JA))=EXPCOE(JB,INNO(JA))                           CNVR
      RETURN                                                            CNVR
      END                                                               CNVR
      SUBROUTINE AITKEN(EXPCOE,ITER,NOBT,ORB)                           AITK
C   AN AITKEN DELTA-SQUARE CONVERGENCE ACCELERATION                     AITK
      IMPLICIT REAL*8(A-H,O-Z)                                          AITK
      REAL*8 EXPCOE(5,10),ARRAY(10,5,3)                                 AITK
      INTEGER NOBT(3),ORB(3),IREPV(10)                                  AITK
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                           AITK
      IF(ITER.NE.1)GOTO2                                                AITK
      NORBT=ORB(1)+ORB(2)+ORB(3)                                        AITK
      DO 1 JA=1,10                                                      AITK
1     IREPV(JA)=0                                                       AITK
2     DO 3 JA=1,NORBT                                                   AITK
      IXP=INNO(JA)                                                      AITK
      ISYP=1+IXP/5+IXP/9-IXP/10                                         AITK
      NOB=NOBT(ISYP)                                                    AITK
      IREPV(IXP)=IREPV(IXP)+1                                           AITK
      DO 4 JB=1,NOB                                                     AITK
4     ARRAY(JA,JB,IREPV(IXP))=EXPCOE(JB,IXP)                            AITK
      IF(IREPV(IXP).NE.3)GOTO3                                          AITK
      DO 6 JB=1,NOB                                                     AITK
      IF(DABS(ARRAY(JA,JB,3)-ARRAY(JA,JB,2)).GE.DABS(ARRAY(JA,JB,2)-ARRAAITK
```

```
     .Y(JA,JB,1)))GOTO7                                                  AITK
6     EXPCOE(JB,IXP)=ARRAY(JA,JB,1)-((ARRAY(JA,JB,2)-ARRAY(JA,JB,1))**2)AITK
     ./(ARRAY(JA,JB,3)-2.D0*ARRAY(JA,JB,2)+ARRAY(JA,JB,1))               AITK
      IREPV(IXP)=0                                                       AITK
      GOTO 3                                                             AITK
7     DO 8 JC=1,NOB                                                      AITK
      ARRAY(JA,JC,1)=ARRAY(JA,JC,2)                                      AITK
8     ARRAY(JA,JC,2)=ARRAY(JA,JC,3)                                      AITK
      IREPV(IXP)=2                                                       AITK
3     CONTINUE                                                           AITK
      RETURN                                                             AITK
      END                                                                AITK
      SUBROUTINE PRPRTS(EXPCOF,ORB,NOBT,ITER,PROSUM,PROPM)               PROP
C   THIS ROUTINE COMPUTES THE EXPECTATION VALUES <1/R>,<R>,<R**2>        PROP
      IMPLICIT REAL*8(A-H,O-Z)                                           PROP
      COMMON/PROPER/SRM1(5,5,3),SRP1(5,5,3),SRP2(5,5,3)                  PROP
      COMMON/ONE/FAC1(50),FHH1(5,5,4),FH1(5,5,4),INT1(50),LIM1           PROP
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                            PROP
      REAL*8 PROSUM(3),PROPM(3,4,3),FACV(10),R1(5),R2(5),R3(5),EXPCOF(5,PROP
     .10)                                                                PROP
      INTEGER ORB(3),NOBT(3)                                             PROP
      IF (ITER.NE.1)GOTO20                                               PROP
      NORBT=ORB(1)+ORB(2)+ORB(3)                                         PROP
      DO 1 JA=1,10                                                       PROP
1     FACV(JA)=0                                                         PROP
      DO 2 JA=1,LIM1                                                     PROP
      IORB=INT1(JA)                                                      PROP
      GOTO(3,4,5,5,5,6,7,7,7,8,8,8,8,8,9,10,10,10,11,11,11,11,11,12,13,1PROP
     .3,13),IORB                                                         PROP
3     IA=1                                                               PROP
      GOTO2                                                              PROP
4     IA=2                                                               PROP
      GOTO2                                                              PROP
5     IA=5                                                               PROP
      GOTO2                                                              PROP
6     IA=3                                                               PROP
      GOTO2                                                              PROP
7     IA=6                                                               PROP
      GOTO2                                                              PROP
8     IA=9                                                               PROP
      GOTO2                                                              PROP
9     IA=4                                                               PROP
      GOTO2                                                              PROP
10    IA=7                                                               PROP
      GOTO2                                                              PROP
11    IA=10                                                              PROP
      GOTO2                                                              PROP
12    IA=11                                                              PROP
      GOTO2                                                              PROP
13    IA=8                                                               PROP
2     FACV(IA)=FACV(IA)+FAC1(JA)                                         PROP
20    DO 15 JA=1,3                                                       PROP
15    PROSUM(JA)=0                                                       PROP
      DO 23 JB=1,3                                                       PROP
      DO 23 JC=1,4                                                       PROP
      DO 23 JD=1,3                                                       PROP
23    PROPM(JB,JC,JD)=0.D0                                               PROP
      DO 21 JA=1,NORBT                                                   PROP
      IXP=INNO(JA)                                                       PROP
      ISYP=1+IXP/5+IXP/9-IXP/10
```

```
      IORB=IXP-(ISYP-1)*4                                                   PROP
      NOB=NOBT(ISYP)                                                        PROP
      DO 22 JB=1,NOB                                                        PROP
      R1(JB)=0.D0                                                           PROP
      R2(JB)=0.D0                                                           PROP
      R3(JB)=0.D0                                                           PROP
      DO 22 JC=1,NOB                                                        PROP
      R1(JB)=R1(JB)+SRM1(JB,JC,ISYP)*EXPCOE(JC,IXP)                         PROP
      R2(JB)=R2(JB)+SRP1(JB,JC,ISYP)*EXPCOE(JC,IXP)                         PROP
22    R3(JB)=R3(JB)+SRP2(JB,JC,ISYP)*EXPCOE(JC,IXP)                         PROP
      DO 24 JB=1,NOB                                                        PROP
      PROPM(ISYP,IORB,1)=PROPM(ISYP,IORB,1)+EXPCOE(JB,IXP)*R1(JB)           PROP
      PROPM(ISYP,IORB,2)=PROPM(ISYP,IORB,2)+EXPCOE(JB,IXP)*R2(JB)           PROP
24    PROPM(ISYP,IORB,3)=PROPM(ISYP,IORB,3)+EXPCOE(JB,IXP)*R3(JB)           PROP
C ADD UP THE PROPERTIES                                                     PROP
      FACT=FACV(IXP)                                                        PROP
      DO 26 JC=1,3                                                          PROP
26    PROSUM(JC)=PROSUM(JC)+PROPM(ISYP,IORB,JC)*FACT                        PROP
21    CONTINUE                                                              PROP
      RETURN                                                                PROP
      END                                                                   PROP
      SUBROUTINE HINZE(EXPCOE,ISYM,ORB,COMPL)                               HINZ
      IMPLICIT REAL*8 (A-H,O-Z)                                             HINZ
      COMMON/HINZ/S,F,L,NOB,NORB,CLOSED                                     HINZ
      COMMON/INTHIN/GSM,GSV                                                 HINZ
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                               HINZ
      REAL*8 EPSI(4,4),GSM(5,5,4,4),GSV(20),D(5,5,4,4),R1(5,5),R2(5,5),RHINZ
     .3(5,5),R4(5,5),RV1(5),RV2(5),S(5,5,3),F(5,5,4),L(5,5,4,4),EXPCOE(5HINZ
     .,10),THRH/1.D-5/                                                      HINZ
C   CLOSED CONTAINS INFO IF THE ORBITAL(ISYM,NORB) BELONG TO CLOSED SHELHINZ
      LOGICAL CLOSED(3,4)                                                   HINZ
      INTEGER ORB(3)                                                        HINZ
C EMPTY ALL ARRAYS                                                          HINZ
      DO 1 JC=1,4                                                           HINZ
      DO 1 JD=1,4                                                           HINZ
      EPSI(JC,JD)=0.D0                                                      HINZ
      DO 1 JA=1,5                                                           HINZ
      DO 1 JB=1,5                                                           HINZ
1     GSM(JA,JB,JC,JD)=0.D0                                                 HINZ
      IST=ISTA(ISYM)                                                        HINZ
5323  CONTINUE                                                              HINZ
C   COMPUTE EPSI(JA,JB)                                                     HINZ
      DO 40 JA=1,NORB                                                       HINZ
      DO 45 JB=JA,NORB                                                      HINZ
      IF(CLOSED(ISYM,JA).AND.CLOSED(ISYM,JB).AND.JA.NE.JB)GOTO45            HINZ
      DO 41 JC=1,NOB                                                        HINZ
      DO 41 JD=1,NOB                                                        HINZ
41    R1(JC,JD)=F(JC,JD,JA)+F(JC,JD,JB)                                     HINZ
      DO 42 JC=1,NOB                                                        HINZ
      RV1(JC)=0.D0                                                          HINZ
      DO 42 JD=1,NOB                                                        HINZ
42    RV1(JC)=RV1(JC)+EXPCOE(JD,INNO(IST+JA))*R1(JD,JC)                     HINZ
      EPSI(JA,JB)=0.D0                                                      HINZ
      DO 43 JC=1,NOB                                                        HINZ
43    EPSI(JA,JB)=RV1(JC)*EXPCOE(JC,INNO(IST+JB))+EPSI(JA,JB)               HINZ
      EPSI(JB,JA)=5.D-1*EPSI(JA,JB)                                         HINZ
      EPSI(JA,JB)=EPSI(JB,JA)                                               HINZ
45    CONTINUE                                                              HINZ
40    CONTINUE                                                              HINZ
      DO 44 JA=1,NORB
```

```
      DO 44 JB=1,NORB                                                   HINZ
      DO 44 JC=1,NOB                                                    HINZ
      DO 44 JD=1,NOB                                                    HINZ
44    D(JC,JD,JA,JB)=EXPCOE(JC,INNO(IST+JA))*EXPCOE(JD,INNO(IST+JB))    HINZ
C   LOOP 5 SETS UP THE G-SUPERMATRIX                                    HINZ
      DO 5 JA=1,NORB                                                    HINZ
      DO 5 JB=1,NORB                                                    HINZ
C THE MATRIX EPSI(I,J)*S-2*L(I,J) IS ADDED                              HINZ
      DO 6 JC=1,NOB                                                     HINZ
      DO 6 JD=1,NOB                                                     HINZ
6     GSM(JC,JD,JA,JB)=GSM(JC,JD,JA,JB)+EPSI(JA,JB)*S(JC,JD,ISYM)-2.D0*LHINZ
     .(JC,JD,JA,JB)                                                     HINZ
      IF(JA.NE.JB)GOTO13                                                HINZ
C   THE PART WHICH CONTRIBUTES ONLY TO THE DIAGONAL ELEMENTS IS DONE    HINZ
      DO 8 JC=1,NORB                                                    HINZ
      IF(CLOSED(ISYM,JA).AND.CLOSED(ISYM,JC).AND.JA.NE.JC)GOTO8         HINZ
      DO 9 JD=1,NOB                                                     HINZ
      DO 9 JE=1,NOB                                                     HINZ
9     R1(JD,JE)=F(JD,JE,JA)+F(JD,JE,JC)                                 HINZ
      DO 10 JD=1,NOB                                                    HINZ
      DO 10 JE=1,NOB                                                    HINZ
      R2(JD,JE)=0.D0                                                    HINZ
      R3(JD,JE)=0.D0                                                    HINZ
      DO 10 JF=1,NOB                                                    HINZ
      R2(JD,JE)=R2(JD,JE)+D(JD,JF,JC,JC)*R1(JF,JE)                      HINZ
10    R3(JD,JE)=R3(JD,JE)+R1(JD,JF)*D(JF,JE,JC,JC)                      HINZ
      DO 11 JD=1,NOB                                                    HINZ
      DO 11 JE=1,NOB                                                    HINZ
      R1(JD,JE)=0.D0                                                    HINZ
      R4(JD,JE)=0.D0                                                    HINZ
      DO 11 JF=1,NOB                                                    HINZ
      R1(JD,JE)=R1(JD,JE)+S(JD,JF,ISYM)*R2(JF,JE)                       HINZ
11    R4(JD,JE)=R4(JD,JE)+R3(JD,JF)*S(JF,JE,ISYM)                       HINZ
      DO 12 JD=1,NOB                                                    HINZ
      DO 12 JE=1,NOB                                                    HINZ
12    GSM(JD,JE,JA,JB)=GSM(JD,JE,JA,JB)+0.5D0*(R1(JD,JE)+R4(JD,JE))     HINZ
8     CONTINUE                                                          HINZ
      DO 121 JC=1,NOB                                                   HINZ
      DO 121 JD=1,NOB                                                   HINZ
121   GSM(JC,JD,JA,JB)=GSM(JC,JD,JA,JB)-F(JC,JD,JA)                     HINZ
C   THIS PART IS ADDED IF ORBITAL(JA) OR (JB) DO NOT BELONG TO          HINZ
C   A CLOSED SHELL                                                      HINZ
13    IF(CLOSED(ISYM,JA).AND.CLOSED(ISYM,JB).AND.JA.NE.JB)GOTO18        HINZ
      DO 14 JC=1,NOB                                                    HINZ
      DO 14 JD=1,NOB                                                    HINZ
14    R1(JC,JD)=F(JC,JD,JA)+F(JC,JD,JB)                                 HINZ
      DO 15 JC=1,NOB                                                    HINZ
      DO 15 JD=1,NOB                                                    HINZ
      R2(JC,JD)=0.D0                                                    HINZ
      R3(JC,JD)=0.D0                                                    HINZ
      DO 15 JE=1,NOB                                                    HINZ
      R2(JC,JD)=R2(JC,JD)+D(JC,JE,JA,JB)*R1(JE,JD)                      HINZ
15    R3(JC,JD)=R3(JC,JD)+R1(JC,JE)*D(JE,JD,JA,JB)                      HINZ
      DO 16 JC=1,NOB                                                    HINZ
      DO 16 JD=1,NOB                                                    HINZ
      R1(JC,JD)=0.D0                                                    HINZ
      R4(JC,JD)=0.D0                                                    HINZ
      DO 16 JE=1,NOB                                                    HINZ
      R1(JC,JD)=R1(JC,JD)+S(JC,JE,ISYM)*R2(JE,JD)                       HINZ
16    R4(JC,JD)=R4(JC,JD)+R3(JC,JE)*S(JE,JD,ISYM)                       HINZ
```

```
      DO 17 JC=1,NOB                                                    HINZ
      DO 17 JD=1,NOB                                                    HINZ
17    GSM(JC,JD,JA,JB)=GSM(JC,JD,JA,JB)+0.5D0*(R1(JC,JD)+R4(JC,JD))     HINZ
C   LOOP 19 ADDS THE L-MATRICES                                         HINZ
18    DO 19 JC=1,NORB                                                   HINZ
      IF(CLOSED(ISYM,JA).AND.CLOSED(ISYM,JC).AND.JA.NE.JC)GOTO23        HINZ
      DO 20 JD=1,NOB                                                    HINZ
      DO 20 JE=1,NOB                                                    HINZ
      R1(JD,JE)=0.D0                                                    HINZ
      R2(JD,JE)=0.D0                                                    HINZ
      DO 20 JF=1,NOB                                                    HINZ
      R1(JD,JE)=R1(JD,JE)+D(JD,JF,JC,JC)*L(JF,JE,JA,JB)                 HINZ
20    R2(JD,JE)=R2(JD,JE)+D(JD,JF,JC,JA)*L(JF,JE,JC,JB)                 HINZ
      DO 21 JD=1,NOB                                                    HINZ
      DO 21 JE=1,NOB                                                    HINZ
21    R3(JD,JE)=R1(JD,JE)+R2(JD,JE)                                     HINZ
      DO 22 JD=1,NOB                                                    HINZ
      DO 22 JE=1,NOB                                                    HINZ
      DO 22 JF=1,NOB                                                    HINZ
22    GSM(JD,JE,JA,JB)=GSM(JD,JE,JA,JB)+S(JD,JF,ISYM)*R3(JF,JE)         HINZ
23    IF(CLOSED(ISYM,JB).AND.CLOSED(ISYM,JC).AND.JB.NE.JC)GOTO19        HINZ
      DO 24 JD=1,NOB                                                    HINZ
      DO 24 JE=1,NOB                                                    HINZ
      R1(JD,JE)=0.D0                                                    HINZ
      R2(JD,JE)=0.D0                                                    HINZ
      DO 24 JF=1,NOB                                                    HINZ
      R1(JD,JE)=R1(JD,JE)+L(JD,JF,JA,JB)*D(JF,JE,JC,JC)                 HINZ
24    R2(JD,JE)=R2(JD,JE)+L(JD,JF,JA,JC)*D(JF,JE,JB,JC)                 HINZ
      DO 25 JD=1,NOB                                                    HINZ
      DO 25 JE=1,NOB                                                    HINZ
25    R3(JD,JE)=R1(JD,JE)+R2(JD,JE)                                     HINZ
      DO 26 JD=1,NOB                                                    HINZ
      DO 26 JE=1,NOB                                                    HINZ
      DO 26 JF=1,NOB                                                    HINZ
26    GSM(JD,JE,JA,JB)=GSM(JD,JE,JA,JB)+R3(JD,JF)*S(JF,JE,ISYM)         HINZ
19    CONTINUE                                                          HINZ
5     CONTINUE                                                          HINZ
C THE G-SUPERVECTOR IS COMPUTED                                         HINZ
      DO 27 JA=1,NORB                                                   HINZ
      DO 28 JB=1,NOB                                                    HINZ
28    RV1(JB)=0.D0                                                      HINZ
      DO 29 JB=1,NORB                                                   HINZ
      DO 29 JC=1,NOB                                                    HINZ
29    RV1(JC)=RV1(JC)+EPSI(JA,JB)*EXPCOE(JC,INNO(IST+JB))               HINZ
      DO 30 JB=1,NOB                                                    HINZ
      RV2(JB)=0.D0                                                      HINZ
      DO 30 JC=1,NOB                                                    HINZ
30    RV2(JB)=RV2(JB)+S(JB,JC,ISYM)*RV1(JC)                             HINZ
      DO 31 JB=1,NOB                                                    HINZ
      RV1(JB)=0.D0                                                      HINZ
      DO 31 JC=1,NOB                                                    HINZ
31    RV1(JB)=RV1(JB)+F(JB,JC,JA)*EXPCOE(JC,INNO(IST+JA))               HINZ
      DO 32 JB=1,NOB                                                    HINZ
32    GSV((JA-1)*5+JB)=RV1(JB)-RV2(JB)                                  HINZ
27    CONTINUE                                                          HINZ
C SUBROUTINE GAUSS AND COMPARISON                                       HINZ
      CALL SOLVER(NOB,NORB,ISYM)                                        HINZ
C   LOOP 33 MAKES A LEAST SQUARE COMPARISON                             HINZ
      DO 33 JA=1,NORB                                                   HINZ
      ITO=(JA-1)*5
```

```
      DO 34 JB=1,NOB                                                    HINZ
34    EXPCOE(JB,INNO(IST+JA))=EXPCOE(JB,INNO(IST+JA))+GSV(ITO+JB)       HINZ
33    CONTINUE                                                          HINZ
      DO 39 JA=1,NORB                                                   HINZ
      DO 39 JB=1,NORB                                                   HINZ
      DO 400 JC=1,NOB                                                   HINZ
      GSV(JC)=0.D0                                                      HINZ
      DO 400 JD=1,NOB                                                   HINZ
400   GSV(JC)=GSV(JC)+EXPCOE(JD,INNO(IST+JA))*S(JD,JC,ISYM)             HINZ
      SUM=0.D0                                                          HINZ
      DO 410 JC=1,NOB                                                   HINZ
410   SUM=SUM+GSV(JC)*EXPCOE(JC,INNO(IST+JB))                           HINZ
      IF(JA.EQ.JB)COMPL=COMPL+DABS(SUM-1.D0)                            HINZ
39    WRITE(10,900)JA,JB,SUM                                            HINZ
900   FORMAT(' ',2I4,D25.16)                                            HINZ
      RETURN                                                            HINZ
      END                                                               HINZ
      SUBROUTINE SOLVER(NOB,NORB,ISYM)                                  SOLV
      IMPLICIT REAL*8(A-H,O-Z)                                          SOLV
      COMMON/INTHIN/GSM(5,5,4,4),GSV(20)                                SOLV
      REAL*8 MAT(400),VEC(20)                                           SOLV
      IDIM=NOB*NORB                                                     SOLV
      JLI=(ISYM-1)*4                                                    SOLV
900   FORMAT(///)                                                       SOLV
901   FORMAT(' ',5D20.10)                                               SOLV
902   FORMAT('1')                                                       SOLV
      DO 1 JA=1,NORB                                                    SOLV
      ICOL=(JA-1)*NOB                                                   SOLV
      DO 1 JB=1,NOB                                                     SOLV
      ICOLB=ICOL+JB                                                     SOLV
      ITO=(ICOLB-1)*IDIM                                                SOLV
      DO 1 JC=1,NORB                                                    SOLV
      IROW=(JC-1)*NOB                                                   SOLV
      DO 1 JD=1,NOB                                                     SOLV
      IROWD=IROW+JD                                                     SOLV
1     MAT(ITO+IROWD)=GSM(JD,JB,JC,JA)                                   SOLV
      DO 2 JA=1,NORB                                                    SOLV
      ITO=(JA-1)*NOB                                                    SOLV
      DO 2 JB=1,NOB                                                     SOLV
2     VEC(ITO+JB)=GSV((JA-1)*5+JB)                                      SOLV
      CALL GAUSS(MAT,VEC,IDIM)                                          SOLV
      DO 3 JA=1,NORB                                                    SOLV
      ITO=(JA-1)*5                                                      SOLV
      ITA=(JA-1)*NOB                                                    SOLV
      DO 3 JB=1,NOB                                                     SOLV
3     GSV(ITO+JB)=VEC(ITA+JB)                                           SOLV
      RETURN                                                            SOLV
      END                                                               SOLV
      SUBROUTINE GAUSS(MAT,VEC,IDIM)                                    GAUS
      IMPLICIT REAL*8 (A-H,O-Z)                                         GAUS
C GAUSS ELIMINATION WITH PIVOTING OF ROWS AND COLUMNS                   GAUS
      REAL*8 MAT(IDIM,IDIM),VEC(IDIM),SOLV(20)                          GAUS
      INTEGER EXVE(20)                                                  GAUS
      IF(IDIM.EQ.1)GOTO20                                               GAUS
      DO 1 JA=1,IDIM                                                    GAUS
1     EXVE(JA)=JA                                                       GAUS
      IDM1=IDIM-1                                                       GAUS
      DO 2 JA=1,IDM1                                                    GAUS
      BMAX=DABS(MAT(JA,JA))                                             GAUS
      IROW=JA                                                           GAUS
```

```
      ICOL=JA
C LOOK FOR LARGEST REMAINING ELEMENT                                    GAUS
      DO 3 JB=JA,IDIM                                                   GAUS
      DO 3 JC=JA,IDIM                                                   GAUS
      IF(DABS(MAT(JB,JC)).LE.BMAX)GOTO3                                 GAUS
      BMAX=DABS(MAT(JB,JC))                                             GAUS
      IROW=JB                                                           GAUS
      ICOL=JC                                                           GAUS
3     CONTINUE                                                          GAUS
C EXCHANGE ROWS(IF NECESSARY)                                           GAUS
      IF(IROW.EQ.JA)GOTO5                                               GAUS
      DO 4 JB=JA,IDIM                                                   GAUS
      EX=MAT(JA,JB)                                                     GAUS
      MAT(JA,JB)=MAT(IROW,JB)                                           GAUS
4     MAT(IROW,JB)=EX                                                   GAUS
      EX=VEC(JA)                                                        GAUS
      VEC(JA)=VEC(IROW)                                                 GAUS
      VEC(IROW)=EX                                                      GAUS
C EXCHANGE COLUMNS AND STORE WHICH HAVE BEEN CHANGED                    GAUS
5     IF(JA.EQ.ICOL)GOTO7                                               GAUS
      IEX=EXVE(JA)                                                      GAUS
      EXVE(JA)=EXVE(ICOL)                                               GAUS
      EXVE(ICOL)=IEX                                                    GAUS
      DO 6 JB=1,IDIM                                                    GAUS
      EX=MAT(JB,JA)                                                     GAUS
      MAT(JB,JA)=MAT(JB,ICOL)                                           GAUS
6     MAT(JB,ICOL)=EX                                                   GAUS
C ELIMINATE JA-TH COLUMN                                                GAUS
7     IS=JA+1                                                           GAUS
      DO 8 JB=IS,IDIM                                                   GAUS
      FAC=-MAT(JB,JA)/MAT(JA,JA)                                        GAUS
      DO 9 JC=JA,IDIM                                                   GAUS
9     MAT(JB,JC)=MAT(JA,JC)*FAC+MAT(JB,JC)                              GAUS
8     VEC(JB)=VEC(JA)*FAC+VEC(JB)                                       GAUS
2     CONTINUE                                                          GAUS
C BACKSUBSTITUTE                                                        GAUS
      SOLV(IDIM)=VEC(IDIM)/MAT(IDIM,IDIM)                               GAUS
      LIM=IDIM-1                                                        GAUS
      DO 10 JA=1,LIM                                                    GAUS
      SUM=VEC(IDIM-JA)                                                  GAUS
      DO 11 JB=1,JA                                                     GAUS
11    SUM=SUM-SOLV(IDIM-JB+1)*MAT(IDIM-JA,IDIM-JB+1)                    GAUS
10    SOLV(IDIM-JA)=SUM/MAT(IDIM-JA,IDIM-JA)                            GAUS
      DO 12 JA=1,IDIM                                                   GAUS
12    VEC(EXVE(JA))=SOLV(JA)                                            GAUS
C CALCULATE THE NORMALIZED DETERMINANT AND CHECK FOR ILLCONDITIONING    GAUS
      ALPHA=1.D0                                                        GAUS
      DO 13 JA=1,IDIM                                                   GAUS
      SOLV(JA)=0.D0                                                     GAUS
      DO 14 JB=JA,IDIM                                                  GAUS
14    SOLV(JA)=SOLV(JA)+MAT(JA,JB)**2                                   GAUS
13    ALPHA=ALPHA*DSQRT(SOLV(JA))                                       GAUS
      SUM=1.D0                                                          GAUS
      DO 15 JA=1,IDIM                                                   GAUS
15    SUM=SUM*MAT(JA,JA)                                                GAUS
      DET=SUM/ALPHA                                                     GAUS
      IF(DABS(DET).GT.1.D-5)RETURN                                      GAUS
      WRITE(6,900)DET                                                   GAUS
      RETURN                                                            GAUS
20    VEC(1)=VEC(1)/MAT(1,1)                                            GAUS
```

```
      RETURN
900   FORMAT(///131('*')/20X,'   THE VALUE OF THE NORM. DETERMINANT IS  GAUS
     .',1PD10.1/131('*'))                                                 GAUS
      END                                                                 GAUS
      SUBROUTINE ENER(ISYM,EXPCOE,FH1,FH2,NOBT,ORB,ENERGY)                GAUS
      IMPLICIT REAL*8(A-H,O-Z)                                            ENER
      INTEGER NOBT(3),ORB(3)                                              ENER
      COMMON/ENRG/VIRIAL(3,4,2)                                           ENER
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                             ENER
      REAL*8 EXPCOE(5,10),FH1(5,5,4),FH2(5,5,4),R(5,5),ENERG(4),RV(5),ENENER
     .ERGY(3,4),RPOT(5),RKIN(5),VKIN(4),VPOT(4)                           ENER
      NORB=ORB(ISYM)                                                      ENER
      NOB=NOBT(ISYM)                                                      ENER
      IST=ISTA(ISYM)                                                      ENER
      DO 1 JA=1,NORB                                                      ENER
      ENERG(JA)=0.D0                                                      ENER
      VPOT(JA)=0.D0                                                       ENER
      VKIN(JA)=0.D0                                                       ENER
      DO 2 JB=1,NOB                                                       ENER
      DO 2 JC=1,NOB                                                       ENER
2     R(JB,JC)=0.5D0*FH2(JB,JC,INNOR(IST+JA))+FH1(JB,JC,INNOR(IST+JA))   ENER
901   FORMAT('0  ENERGY= ',5D20.10/)                                      ENER
C   MULTIPLY THE RESULTANT MATRIX BY TH E-VECTOR                          ENER
C   COMPUTE ALSO THE TERMS CONTRIBUTING TO THE POTENTIAL AND KINETIC ENEENER
      DO 3 JB=1,NOB                                                       ENER
      RV(JB)=0.D0                                                         ENER
      RPOT(JB)=0.D0                                                       ENER
      RKIN(JB)=0.D0                                                       ENER
      DO 3 JC=1,NOB                                                       ENER
      RKIN(JB)=RKIN(JB)+EXPCOE(JC,INNO(IST+JA))*FH1(JC,JB,INNOR(IST+JA))ENER
      RPOT(JB)=RPOT(JB)+EXPCOE(JC,INNO(IST+JA))*FH2(JC,JB,INNOR(IST+JA))ENER
3     RV(JB)=RV(JB)+EXPCOE(JC,INNO(IST+JA))*R(JC,JB)                      ENER
      DO 5 JB=1,NOB                                                       ENER
      VPOT(JA)=VPOT(JA)+RPOT(JB)*EXPCOE(JB,INNO(IST+JA))                  ENER
      VKIN(JA)=VKIN(JA)+RKIN(JB)*EXPCOE(JB,INNO(IST+JA))                  ENER
5     ENERG(JA)=ENERG(JA)+RV(JB)*EXPCOE(JB,INNO(IST+JA))                  ENER
1     CONTINUE                                                            ENER
      EN=0.D0                                                             ENER
      DO 4 JA=1,NORB                                                      ENER
      VIRIAL(ISYM,JA,1)=VPOT(JA)*0.5D0                                    ENER
      VIRIAL(ISYM,JA,2)=VKIN(JA)                                          ENER
      ENERGY(ISYM,JA)=ENERG(JA)                                           ENER
4     EN=EN+ENERG(JA)                                                     ENER
      WRITE(8,901)EN,(ENERG(JA),JA=1,NORB)                                ENER
      RETURN                                                              ENER
      END                                                                 ENER
      SUBROUTINE EXVAHH(FHH1,FHH2,FHH3,FHH4,EXPCOE,ISYM,NOBT,ORB,EXHH)    EXHH
      IMPLICIT REAL*8(A-H,O-Z)                                            EXHH
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                             EXHH
      REAL*8 FHH1(5,5,4),FHH2(5,5,4),FHH3(5,5,4),FHH4(5,5,4),EXPCOE(5,10EXHH
     .),R(5,5),RV(5),EHH(4),EXHH(3,4)                                     EXHH
      INTEGER NOBT(3),ORB(3)                                              EXHH
      IST=ISTA(ISYM)                                                      EXHH
      NORB=ORB(ISYM)                                                      EXHH
      NOB=NOBT(ISYM)                                                      EXHH
      DO 1 JA=1,NORB                                                      EXHH
      JD=INNOR(IST+JA)                                                    EXHH
      DO 2 JB=1,NOB                                                       EXHH
      DO 2 JC=1,NOB                                                       EXHH
2     R(JB,JC)=FHH1(JB,JC,JD)+0.5D0*FHH2(JB,JC,JD)+FHH3(JB,JC,JD)/3.D0+  EXHH
```

```
     .0.25D0*FHH4(JB,JC,JD)                                            EXHH
      EHH(JA)=0.D0                                                      EXHH
      DO 4 JB=1,NOB                                                     EXHH
      RV(JB)=0.D0                                                       EXHH
      DO 4 JC=1,NOB                                                     EXHH
4     RV(JB)=RV(JB)+EXPCOE(JC,INNO(IST+JA))*R(JC,JB)                    EXHH
      DO 1 JB=1,NOB                                                     EXHH
      EHH(JA)=EHH(JA)+RV(JB)*EXPCOE(JB,INNO(IST+JA))                    EXHH
1     EXHH(ISYM,JA)=EHH(JA)                                             EXHH
      EN2=0.D0                                                          EXHH
      DO 5 JA=1,NORB                                                    EXHH
5     EN2=EN2+EHH(JA)                                                   EXHH
      WRITE(8,900)EN2,(EHH(JA),JA=1,NORB)                               EXHH
900   FORMAT('0  EVHH  = ',5D20.10/)                                    EXHH
      RETURN                                                            EXHH
      END                                                               EXHH
      SUBROUTINE OPTIM(ORB,ENERGY,EXHH,WK,METHOD,IOPTI,LIM4,TAU)        OPTI
      IMPLICIT REAL*8(A-H,O-Z)                                          OPTI
      COMMON/ALL/EXPCOE(5,10),ORBEXP(15),H(5,5,3),CHARGE,QN,NOBT(3),ISYMOPTI
     .,FDUB                                                             OPTI
      REAL*8 ENERGY(3,4),EXHH(3,4),XVEC(10),YVEC(10),MAT(10,10),VEC(10) OPTI
      INTEGER QN(15),FDUB,ORB(3),NOEXV(15)                              OPTI
C   LOOP 1 RUNS OVER THE SYMMETRIES                                     OPTI
      READ(5,901)NOE,(NOEXV(JA),JA=1,NOE),ICHNGE                        OPTI
901   FORMAT(20I4)                                                      OPTI
      DO 1 JA=1,NOE                                                     OPTI
      NOEX=NOEXV(JA)                                                    OPTI
C   LOOP 2 RUNS OVER THE BASIS FUNCTIONS                                OPTI
      ICOND=0                                                           OPTI
      CALL SCFCYC(ORB,ENERGY,EXHH,WK,LIM4,METHOD,TAU)                   OPTI
      XVAL=ORBEXP(NOEX)                                                 OPTI
      DO 3 JC=1,10                                                      OPTI
C   LOOPS 4&5 ADD UP THE ORBITAL ENERGIES                               OPTI
      EVH=0.D0                                                          OPTI
      EVHH=0.D0                                                         OPTI
      DO 4 JD=1,3                                                       OPTI
      NEWNOR=ORB(JD)                                                    OPTI
      IF(NEWNOR.EQ.0)GOTO4                                              OPTI
      DO 5 JE=1,NEWNOR                                                  OPTI
      EVH=EVH+ENERGY(JD,JE)                                             OPTI
5     EVHH=EVHH+EXHH(JD,JE)                                             OPTI
4     CONTINUE                                                          OPTI
      GO TO(7,8,9,10,11),METHOD                                         OPTI
7     YVAL=EVH                                                          OPTI
      GOTO12                                                            OPTI
8     YVAL=EVHH-EVH*EVH                                                 OPTI
      GOTO12                                                            OPTI
9     YVAL=EVHH-2.D0*EVH*WK+WK*WK                                       OPTI
      GOTO12                                                            OPTI
10    YVAL=((EVH-WK)**2)/(EVHH-2.D0*EVH*WK+WK*WK)                       OPTI
      GOTO12                                                            OPTI
11    YVAL=((EVH-WK)**2)/(EVHH-EVH*EVH)                                 OPTI
12    IF(ICOND.EQ.1)GOTO6                                               OPTI
      CALL CHANGE(XVAL,YVAL,XVEC,YVEC,JC,ICOND,ICHNGE)                  OPTI
      ORBEXP(NOEX)=XVAL                                                 OPTI
      CALL REWIND(2)                                                    OPTI
3     CALL SCFCYC(ORB,ENERGY,EXHH,WK,LIM4,METHOD,TAU)                   OPTI
      WRITE(6,900)                                                      OPTI
900   FORMAT('  THE TEN POINTS IN OPTIM ARE NOT INCLUDING A MINIMUM')   OPTI
      STOP
```

```
6     YVEC(JC)=YVAL                                                     OPTI
99    CALL POLYNO(XVEC,YVEC,MAT,VEC,JC)                                 OPTI
      ORBEXP(NOEX)=XVEC(1)                                              OPTI
2     CONTINUE                                                          OPTI
1     CONTINUE                                                          OPTI
      RETURN                                                            OPTI
      DEBUG UNIT(9),SUBCHK,SUBTRACE,INIT(ORBEXP,YVAL,XVAL)              OPTI
      AT99                                                              OPTI
      DISPLAY XVEC,YVEC                                                 OPTI
      END                                                               OPTI
      SUBROUTINE SCFCYC(ORB,ENERGY,EXHH,WK,LIM4,METHOD,TAU)             SCFC
      IMPLICIT REAL*8 (A-H,O-Z)                                         SCFC
      COMMON/ALL/EXPCOE(5,10),ORBEXP(15),H(5,5,3),CHARGE,QN,NOBT(3),ISYMSCFC
     .,FDUB                                                             SCFC
      INTEGER  INTNO2,QN(15),FDUB,INFO(4),ORB(3)                        SCFC
      COMMON/HINZ/S,F,L,NOB,NORB,CLOSED                                 SCFC
      COMMON/ONE/FAC1(50),FHH1(5,5,4),FH1(5,5,4),INT1(50),LIM1          SCFC
      COMMON/TWO/FH2(5,5,4),FHH2(5,5,4),FAC2(100),LH2(5,5,4,4),LHH2(5,5,SCFC
     .4,4),INT2(4,100),INTNO2(100),NULL2(100),LIM2,INTLI2               SCFC
      COMMON/THREE/FAC3(200),FHH3(5,5,4),LHH3(5,5,4,4),INT3(6,200),INTNOSCFC
     .3(3,100),LIM3,INTLI3,NULL3                                        SCFC
      REAL*8 F(5,5,4),LH2,LHH2,S(5,5,3),HH(5,5,3),L(5,5,4,4),OENER(3,4),SCFC
     .LHH3,LHH4(5,5,4,4),FHH3,FHH4(5,5,4),EXHH(3,4),ENERGY(3,4)         SCFC
      LOGICAL NULL2                                                     SCFC
      WRITE(11,801)LIM3,LIM4,METHOD                                     SCFC
801   FORMAT(' L3,L4,METHOD= ',3I4)                                     SCFC
      CALL ONEINT(HH,S)                                                 SCFC
      INTLI2=0                                                          SCFC
      INTLI3=0                                                          SCFC
      DO 10 JA=1,3                                                      SCFC
      IF(ORB(JA).EQ.0)GOTO10                                            SCFC
      CALL RENORM(NOBT(JA),ORB(JA),JA,EXPCOE,S)                         SCFC
10    CONTINUE                                                          SCFC
      DO 20 ITER=1,10                                                   SCFC
      LIMDI3=0                                                          SCFC
      LIMDI4=0                                                          SCFC
      COMPL=0.D0                                                        SCFC
      WRITE(8,900)ITER                                                  SCFC
      DO 21 ISYM=1,3                                                    SCFC
      NORB=ORB(ISYM)                                                    SCFC
      NOB=NOBT(ISYM)                                                    SCFC
      IF(NORB.EQ.0)GOTO21                                               SCFC
      DO 30 JA=1,NOB                                                    SCFC
      DO 30 JB=1,NOB                                                    SCFC
      DO 30 JC=1,4                                                      SCFC
      FH1(JA,JB,JC)=0.D0                                                SCFC
      FHH1(JA,JB,JC)=0.D0                                               SCFC
      FH2(JA,JB,JC)=0.D0                                                SCFC
      FHH2(JA,JB,JC)=0.D0                                               SCFC
      FHH3(JA,JB,JC)=0.D0                                               SCFC
      FHH4(JA,JB,JC)=0.D0                                               SCFC
      DO 30 JD=1,4                                                      SCFC
      LH2(JA,JB,JC,JD)=0.D0                                             SCFC
      LHH2(JA,JB,JC,JD)=0.D0                                            SCFC
      LHH3(JA,JB,JC,JD)=0.D0                                            SCFC
30    LHH4(JA,JB,JC,JD)=0.D0                                            SCFC
      CALL ONEEL(NOB,ISYM,H,HH)                                         SCFC
      CALL TIME(1,1)                                                    SCFC
      CALL TWOELE                                                       SCFC
      IF(METHOD.EQ.1.OR.LIM3.EQ.0)GOTO31
```

```
      CALL THREFL(LIMDI3,NORB,NOB)                                      SCFC
      IF(LIM4.EQ.0)GOTO31                                               SCFC
      CALL FOUREL(NORB,NOB,LIMDI4)                                      SCFC
31    CALL COMBIN(METHOD,ISYM,ORB,NORT,FH1,FHH1,FH2,FHH2,FHH3,FHH4,LH2,LSCFC
     .HH2,LHH3,LHH4,WK,EXPCOF,ENERGY,EXHH,TAU)                          SCFC
      CALL HINZE(EXPCOE,ISYM,ORB,COMPL)                                 SCFC
      WRITE(8,902)COMPL                                                 SCFC
902   FORMAT('0  COMPL= ',1PD8.1)                                       SCFC
      CALL RENORM(NOB,NORB,ISYM,EXPCOF,S)                               SCFC
      CALL ENER(ISYM,EXPCOF,FH1,FH2,NORT,ORB,ENERGY)                    SCFC
      CALL EXVAHH(FHH1,FHH2,FHH3,FHH4,EXPCOF,ISYM,NORT,ORB,EXHH)        SCFC
      CALL OUT01(EXPCOF,NOB,NORB,ISYM)                                  SCFC
21    CONTINUE                                                          SCFC
      CALL OUTPUT(EXPCOE,ORBEXP,EXHH,ENERGY,WK,COMPL,ORB,NORT,METHOD,ITESCFC
     .R,QN,ICOMPL,CHARGE)                                               SCFC
      IF(COMPL.LT.1.D-10)RETURN                                         SCFC
      CALL CNVRGC(EXPCOF,ITER,NORT,ORB,&23)                             SCFC
      CALL AITKEN(EXPCOE,ITER,NORT,ORB)                                 SCFC
20    CALL REWIND(3)                                                    SCFC
      RETURN                                                            SCFC
100   WRITE(6,901)                                                      SCFC
22    STOP                                                              SCFC
23    RETURN                                                            SCFC
900   FORMAT(///' ITERATION NO. ',I3)                                   SCFC
901   FORMAT(' LOGIOU HAS WRONG RETURN')                                SCFC
      DEBUG UNIT(9),INIT(COMPL),SUBCHK,SUBTRACE                         SCFC
      END                                                               SCFC
      SUBROUTINE CHANGE(XVAL,YVAL,XVEC,YVEC,NOM,JCOND,ICHNGE)           CHNG
      REAL*8 DELTA,XVAL,YVAL,XVEC(10),YVEC(10)                          CHNG
      JCOND=0                                                           CHNG
      XVEC(NOM)=XVAL                                                    CHNG
      YVEC(NOM)=YVAL                                                    CHNG
      IF(NOM-2)1,2,3                                                    CHNG
1     DELTA=XVAL/DFLOAT(ICHNGE)                                         CHNG
      XVAL=XVAL+DELTA                                                   CHNG
      RETURN                                                            CHNG
2     DELTA=DELTA+DELTA                                                 CHNG
      XVAL=XVAL+DELTA                                                   CHNG
      IF(YVEC(2).LT.YVEC(1))RETURN                                      CHNG
      DELTA=-5.D-1*DELTA                                                CHNG
      XVAL=XVEC(1)+DELTA                                                CHNG
      RETURN                                                            CHNG
3     IF(NOM.GT.3)GOTO4                                                 CHNG
      IF(YVEC(3).LT.YVEC(1))GOTO5                                       CHNG
      DELTA=-5.D-1*DELTA                                                CHNG
      ICOND=1                                                           CHNG
      XVAL=XVAL+DELTA                                                   CHNG
      RETURN                                                            CHNG
5     DELTA=DELTA+DELTA                                                 CHNG
      XVAL=XVAL+DELTA                                                   CHNG
      ICOND=0                                                           CHNG
      RETURN                                                            CHNG
4     IF(ICOND.EQ.0)GOTO6                                               CHNG
      XVAL=XVAL+1.5D0*DELTA                                             CHNG
      XVEC(NOM+1)=XVAL                                                  CHNG
      JCOND=1                                                           CHNG
      RETURN                                                            CHNG
6     IF(YVEC(NOM).LT.YVEC(NOM-1))GOTO7                                 CHNG
      ICOND=1                                                           CHNG
      DELTA=-5.D-1*DELTA
```

```
      XVAL=XVAL+DELTA
      RETURN                                                            CHNG
7     DELTA=DELTA+DELTA                                                 CHNG
      XVAL=XVAL+DELTA                                                   CHNG
      RETURN                                                            CHNG
      DEBUG UNIT(9),SUBCHK                                              CHNG
      END                                                               CHNG
      SUBROUTINE POLYNO(XVEC,YVEC,MAT,VEC,NOPOI)                        CHNG
      IMPLICIT REAL*8(A-H,O-Z)                                          POLY
      REAL*8 XVEC(10),YVEC(10),MAT(NOPOI,NOPOI),VEC(NOPOI),XP(9),XPP(9) POLY
C   OBTAIN THE COEFFICIENTS OF THE APPROXIMATING POLYNOMIAL             POLY
      XNEW=XVEC(1)                                                      POLY
      EX=YVEC(1)                                                        POLY
      DO 6 JA=2,NOPOI                                                   POLY
      IF(EX.LT.YVEC(JA))GOTO6                                           POLY
      EX=YVEC(JA)                                                       POLY
      XNEW=XVEC(JA)                                                     POLY
6     CONTINUE                                                          POLY
      LIM1=NOPOI-1                                                      POLY
      LIM2=NOPOI-2                                                      POLY
      DO 1 JA=1,NOPOI                                                   POLY
      VEC(JA)=YVEC(JA)                                                  POLY
      MAT(JA,1)=1                                                       POLY
      DO 1 JB=1,LIM1                                                    POLY
1     MAT(JA,JB+1)=XVEC(JA)**JB                                         POLY
99    CALL GAUSS(MAT,VEC,NOPOI)                                         POLY
C   DIFFERENTIATE THE APPROXIMATING POLYNOMIAL                          POLY
      DO 2 JA=1,LIM1                                                    POLY
2     XP(JA)=VEC(JA+1)*JA                                               POLY
C   SOLVE FOR THE ZERO BY NEWTONS METHOD                                POLY
C  1: FORM THE DERIVATIVE                                               POLY
      DO 3 JA=1,LIM2                                                    POLY
3     XPP(JA)=XP(JA+1)*JA                                               POLY
C  2: EVALUATE FX AND FPX                                               POLY
999   CONTINUE                                                          POLY
4     XOLD=XNEW                                                         POLY
      FX=0.D0                                                           POLY
      FPX=0.D0                                                          POLY
      DO 5 JA=1,LIM2                                                    POLY
      FX=FX*XOLD+XP(NOPOI-JA)                                           POLY
5     FPX=FPX*XOLD+XPP(LIM1-JA)                                         POLY
      FX=FX*XOLD+XP(1)                                                  POLY
      DELTA=FX/FPX                                                      POLY
      XNEW=XOLD-DELTA                                                   POLY
      IF(1.D9.LT.DABS(DELTA))XNEW=XOLD+1.D0                             POLY
      IF(1.D-6.LT.DABS(XOLD-XNEW))GOTO4                                 POLY
      XVEC(1)=XNEW                                                      POLY
      RETURN                                                            POLY
      DEBUG UNIT(9),SUBTRACE,SUBCHK,INIT(XNEW)                          POLY
      AT 999                                                            POLY
      DISPLAY XP,XPP                                                    POLY
      END                                                               POLY
      SUBROUTINE OUTPUT(EXPCOE,ORBEXP,EXHH,ENERGY,WK,COMPL,ORB,NORT,METHOUT2
     .OD,ITER,QN,ICOMPL,CHARGE)                                         OUT2
      IMPLICIT REAL*8(A-H,O-Z)                                          OUT2
      COMMON/RENOR/INNO(10),ISTA(3),INNOR(10)                           OUT2
      COMMON/ENRG/VIRIAL(3,4,2)                                         OUT2
      REAL*8 EXPCOE(5,10),ORBEXP(15),EXHH(3,4),ENERGY(3,4),VEC(5)       OUT2
      REAL*8 PROPM(3,4,3),PROSUM(3),CUSP(3,4)                           OUT2
      INTEGER ORB(3),NOBT(3),HEAD1(3)/' S  ',' P  ',' D  '/,BLANK/'    'OUT2
```

```
     ./,HEAD2(10)/'1S- ','2S- ','3S- ','4S- ','2P- ','3P- ','4P- ','5P- OUT2
     .','3D- ','4D- '/,HEAD3(3)/'BASI','S/OR','R F '/,HEAD4(3)/'  OR',   OUT2
     .'BITA','L   '/,BLANKL(32)/32*'    '/                              OUT2
      INTEGER QN(15),LINE(20),LINE1(32),LINE2(32)                       OUT2
      IF(ITER.EQ.1.AND.ICOMPL.EQ.0)READ(5,919)(LINE(JA),JA=1,20)        OUT2
      WRITE(6,920)(LINE(JA),JA=1,20)                                    OUT2
      IF(METHOD.EQ.1)WRITE(6,900)                                       OUT2
      IF(METHOD.EQ.2)WRITE(6,901)                                       OUT2
      IF(METHOD.EQ.3)WRITE(6,902)                                       OUT2
      IF(METHOD.EQ.4)WRITE(6,903)                                       OUT2
      IF(METHOD.EQ.5)WRITE(6,923)                                       OUT2
      WRITE(6,921)ITER                                                  OUT2
      NORBT=ORB(1)+ORB(2)+ORB(3)                                        OUT2
C   LOOP 20 COMPUTES THE CUSP FOR EACH ORBITAL                          OUT2
      DO 20 JA=1,NORBT                                                  OUT2
      ISYP=1+INNO(JA)/5+INNO(JA)/9-INNO(JA)/10                          OUT2
      NOB=NOBT(ISYP)                                                    OUT2
      IST=(ISYP-1)*5                                                    OUT2
      SUMNUM=0.D0                                                       OUT2
      SUMDEN=0.D0                                                       OUT2
      DO 21 JB=1,NOB                                                    OUT2
      IF(QN(IST+JB).NE.ISYP+1)GOTO22                                    OUT2
      ENM1=ENMI(ISYP+1,ISYP,ISYP,ORBEXP(IST+JB))                        OUT2
      SUMNUM=SUMNUM+EXPCOE(JB,INNO(JA))*ENM1                            OUT2
      GOTO21                                                            OUT2
22    IF(QN(IST+JB).NE.ISYP)GOTO21                                      OUT2
      ENM2=ENMI(ISYP,ISYP-1,ISYP-1,ORBEXP(JB+IST))                      OUT2
      SUMNUM=SUMNUM-ORBEXP(JB+IST)*EXPCOE(JB,INNO(JA))*ENM2             OUT2
      SUMDEN=SUMDEN+EXPCOE(JB,INNO(JA))*ENM2                            OUT2
21    CONTINUE                                                          OUT2
      JX=INNO(JA)-(ISYP-1)*4                                            OUT2
      CUSP(ISYP,JX)=99999.99D0                                          OUT2
      IF(SUMDEN.NE.0.D0)CUSP(ISYP,JX)=SUMNUM/SUMDEN                     OUT2
20    CONTINUE                                                          OUT2
C   THE TOTAL ENERGY AND TOTAL EXHH IS COMPUTED                         OUT2
      CALL PRPRTS(EXPCOE,ORB,NOBT,ITER,PROSUM,PRORM)                    OUT2
      VIRN=0.D0                                                         OUT2
      VIRD=0.D0                                                         OUT2
      TOTEN=0.D0                                                        OUT2
      TOTEHH=0.D0                                                       OUT2
      DO 1 JA=1,3                                                       OUT2
      LIM=ORB(JA)                                                       OUT2
      IF(LIM.EQ.0)GOTO1                                                 OUT2
      DO 2 JB=1,LIM                                                     OUT2
      VIRN=VIRN+VIRIAL(JA,JB,1)                                         OUT2
      VIRD=VIRD+VIRIAL(JA,JB,2)                                         OUT2
      TOTEN=TOTEN+ENERGY(JA,JB)                                         OUT2
2     TOTEHH=TOTEHH+EXHH(JA,JB)                                         OUT2
1     CONTINUE                                                          OUT2
      VIRIAT=(VIRN-CHARGE*PROSUM(1))/(VIRD+CHARGE*PROSUM(1))            OUT2
      EPSILO=(WK-TOTEN)                                                 OUT2
      DELTAS=TOTEHH+WK*WK-2.D0*TOTEN*WK                                 OUT2
      DELTA=TOTEHH-TOTEN*TOTEN                                          OUT2
      WRITE(6,904)COMPL,TOTEN,TOTEHH,WK,EPSILO,DELTA,DELTAS,(PROSUM(JC),OUT2
     .JC=1,3),VIRIAT                                                    OUT2
C   THE INDIVIDUAL ORBITAL ENERGIES ARE WRITTEN OUT                     OUT2
      WRITE(6,905)                                                      OUT2
      DO 3 JA=1,3                                                       OUT2
      LIM=ORB(JA)                                                       OUT2
      IF(LIM.EQ.0)GOTO3
```

```
      IST=ISTA(JA)
      DO 4 JB=1,LIM                                                      OUT2
      IXP=INNO(IST+JB)                                                   OUT2
      IORB=IXP-(JA-1)*4                                                  OUT2
      IC=HEAD2(IXP)                                                      OUT2
4     WRITE(6,906)IC,ENERGY(JA,JB),EXHH(JA,JB),(PROPM(JA,IORB,JC),JC=1,3OUT2
     .),CUSP(JA,IORB)                                                    OUT2
3     CONTINUE                                                           OUT2
C   THE BASISFUNCTIONS AND VECTORS ARE WRITTEN OUT                       OUT2
      DO 5 JA=1,32                                                       OUT2
      LINE1(JA)=BLANK                                                    OUT2
5     LINE2(JA)=BLANK                                                    OUT2
C   LOOP 6 SETS UP THE HEADINGS                                          OUT2
      IS=0                                                               OUT2
      DO 6 JA=1,3                                                        OUT2
      LIM=ORB(JA)                                                        OUT2
      IF(LIM.EQ.0)GOTO6                                                  OUT2
      IST=ISTA(JA)                                                       OUT2
      IS=IS+3                                                            OUT2
      LINE1(IS)=HEAD1(JA)                                                OUT2
      DO 7 JB=1,3                                                        OUT2
7     LINE2(IS-2+JB)=HEAD3(JB)                                           OUT2
      DO 8 JB=1,LIM                                                      OUT2
      IS=IS+3                                                            OUT2
      LINE1(IS)=HEAD2(INNO(IST+JB))                                      OUT2
      LINE2(IS-1)=HEAD4(1)                                               OUT2
      LINE2(IS)=HEAD4(2)                                                 OUT2
8     LINE2(IS+1)=HEAD4(3)                                               OUT2
6     CONTINUE                                                           OUT2
      WRITE(6,922)                                                       OUT2
      WRITE(6,907)(LINE1(JA),JA=1,32)                                    OUT2
      WRITE(6,907)(LINE2(JA),JA=1,32)                                    OUT2
      MAX=0                                                              OUT2
      DO 9 JA=1,3                                                        OUT2
9     MAX=MAX0(NOBT(JA),MAX)                                             OUT2
      ISI1=0                                                             OUT2
      ISI2=0                                                             OUT2
      IF(ORB(1).NE.0)ISI1=1                                              OUT2
      IF(ORB(2).NE.0)ISI2=ISI1+1                                         OUT2
      IBLAN1=ISI1*16+ORB(1)*12                                           OUT2
      IBLAN2=ISI2*16+ORB(2)*12+IBLAN1                                    OUT2
      DO 10 JA=1,MAX                                                     OUT2
      DO 11 JB=1,3                                                       OUT2
      LIM=ORB(JB)                                                        OUT2
      IF(LIM.EQ.0)GOTO11                                                 OUT2
      IST=ISTA(JB)                                                       OUT2
      IF(JA.GT.NOBT(JB))GOTO11                                           OUT2
      IS=(JB-1)*5                                                        OUT2
      IQN1=QN(IS+JA)                                                     OUT2
      VEC(1)=ORBEXP(IS+JA)                                               OUT2
      DO 12 JC=1,LIM                                                     OUT2
12    VEC(JC+1)=EXPCOE(JA,INNO(IST+JC))                                  OUT2
      LIM1=LIM+1                                                         OUT2
      IF(JB.GT.1)GOTO13                                                  OUT2
      WRITE(6,908)IQN1,(VEC(JC),JC=1,LIM1)                               OUT2
      GOTO11                                                             OUT2
13    IBLAN=IBLAN1                                                       OUT2
      IF(JB.EQ.3)IBLAN=IBLAN2                                            OUT2
      IF(IBLAN.EQ.28)WRITE(6,909)IQN1,(VEC(JC),JC=1,LIM1)                OUT2
      IF(IBLAN.EQ.40)WRITE(6,910)IQN1,(VEC(JC),JC=1,LIM1)
```

```
      IF(IBLAN.EQ.52)WRITE(6,911)IQN1,(VEC(JC),JC=1,LIM1)
      IF(IBLAN.EQ.64)WRITE(6,912)IQN1,(VEC(JC),JC=1,LIM1)               OUT2
      IF(IBLAN.EQ.56)WRITE(6,913)IQN1,(VEC(JC),JC=1,LIM1)               OUT2
      IF(IBLAN.EQ.68)WRITE(6,914)IQN1,(VEC(JC),JC=1,LIM1)               OUT2
      IF(IBLAN.EQ.80)WRITE(6,915)IQN1,(VEC(JC),JC=1,LIM1)               OUT2
      IF(IBLAN.EQ.92)WRITE(6,916)IQN1,(VEC(JC),JC=1,LIM1)               OUT2
      IF(IBLAN.EQ.104)WRITE(6,917)IQN1,(VEC(JC),JC=1,LIM1)              OUT2
      IF(IBLAN.LE.104)GOTO11                                            OUT2
      WRITE(6,918)                                                      OUT2
      STOP                                                              OUT2
11    CONTINUE                                                          OUT2
10    CONTINUE                                                          OUT2
      RETURN                                                            OUT2
900   FORMAT(//19X,' <H>-MINIMIZATION')                                 OUT2
901   FORMAT(//19X,' <(H-E)**2>-MINIMIZATION')                          OUT2
902   FORMAT(//19X,' <(H-WK)**2>-MINIMIZATION')                         OUT2
903   FORMAT(//20X,'<H-WK>**2/<(H-WK)**2>-MINIMIZATION')                OUT2
904   FORMAT(//20X,'COMPLETION',8X,1PD20.2/20X,'<H>',15X,0PF20.10/20X,'<OUT2
     .H**2>',12X,F20.10/20X,'WK',16X,F15.5/20X,'EPSILON',11X,F20.10/20X,OUT2
     .'DELTA',13X,F20.10/20X,'DELTA-TILDE',7X,F20.10/20X,'<1/R>',13X,F20OUT2
     ..10/20X,'<R>',15X,F20.10/20X,'<R**2>',12X,F20.10/20X,'VIRIAL-THM',OUT2
     .8X,F20.10//)                                                      OUT2
906   FORMAT(/4X,A3,'ORBITAL',7(1X,F14.9))                              OUT2
905   FORMAT(21X,'<H>',11X,'<H**2>',10X,'<1/R>',9X,'<R>',11X,'<R**2>',10OUT2
     .X,'CUSP')                                                         OUT2
907   FORMAT(32A4)                                                      OUT2
908   FORMAT(5X,I2,F7.3,F12.8,3F12.8)                                   OUT2
909   FORMAT('+',28X,I2,F7.3,F12.8,3F12.8)                              OUT2
910   FORMAT('+',40X,I2,F7.3,F12.8,3F12.8)                              OUT2
911   FORMAT('+',52X,I2,F7.3,F12.8,3F12.8)                              OUT2
912   FORMAT('+',64X,I2,F7.3,F12.8,3F12.8)                              OUT2
913   FORMAT('+',56X,I2,F7.3,F12.8,3F12.8)                              OUT2
914   FORMAT('+',68X,I2,F7.3,F12.8,3F12.8)                              OUT2
915   FORMAT('+',80X,I2,F7.3,F12.8,3F12.8)                              OUT2
916   FORMAT('+',92X,I2,F7.3,F12.8,3F12.8)                              OUT2
917   FORMAT('+',104X,I2,F7.3,F12.8,3F12.8)                             OUT2
918   FORMAT('   THE LENGTH OF THE LINE IN OUTPUT HAS BEEN EXCEEDED.ERROOUT2
     .R.')                                                              OUT2
919   FORMAT(20A4)                                                      OUT2
920   FORMAT('1'//20X,20A4/)                                            OUT2
921   FORMAT(//20X,'THE RESULTS AFTER THE',I3,'. ITERATION ARE:')       OUT2
922   FORMAT(///)                                                       OUT2
923   FORMAT(//19X,' <H-WK>**2/<(H-E)**2>-MINIMIZATION')                OUT2
      END                                                               OUT2
      SUBROUTINE SPLIT2(NOBT,ISY1B,ISY2B)                               SPLI
      COMMON/SPLI1/I1,I2,J1,J2,K1,K2,L1,L2,LIMI,LIMJ,LIMK,LIML,JMI,JMJ,JSPLI
     .MK,JML,IEXP,JEXP,KEXP,LEXP                                        SPLI
      COMMON/SYM/IDAR(8,10)                                             SPLI
      INTEGER NOBT(3),IS1(4),IS2(4),IS3(8)                              SPLI
      EQUIVALENCE(IDAR(9),IS1(1)),(IDAR(13),IS2(1))                     SPLI
      EQUIVALENCE(I1,IS3(1)),(I2,IS3(2)),(J1,IS3(3)),(J2,IS3(4)),(K1,IS3SPLI
     .(5)),(K2,IS3(6)),(L1,IS3(7)),(L2,IS3(8))                          SPLI
      IJN1(I,J)=MIN0(I,J)+MAX0(I,J)*(MAX0(I,J)-1)/2                     SPLI
      DO 1 JA=1,8                                                       SPLI
1     IDAR(JA,9)=IS3(JA)                                                SPLI
      CALL SYMASI(1,I1,LIMI,L1B,M1B,JMI,NOBT,ISY1B)                     SPLI
      IEXP=(ISY1B-1)*4+JMI                                              SPLI
      CALL SYMASI(1,J1,LIMJ,L2B,M2B,JMJ,NOBT,ISY2B)                     SPLI
      JEXP=(ISY2B-1)*4+JMJ                                              SPLI
      CALL IDNOM(I1,I2,J1,J2,IS1)
```

```
      CALL SYMAS2(NOBT,4,IS3)
      RETURN                                                            SPLI
      ENTRY SPLIT3(IND,NOBT,ISY1B,ISY2B,ISY3B)                          SPLI
      DO 2 JA=1,8                                                       SPLI
2     IDAR(JA,9)=IS3(JA)                                                SPLI
      CALL SYMASI(1,I1,LIMI,L1B,M1B,JMI,NOBT,ISY1B)                     SPLI
      IEXP=(ISY1B-1)*4+JMI                                              SPLI
      CALL SYMASI(1,J1,LIMJ,L2B,M2B,JMJ,NOBT,ISY2B)                     SPLI
      JEXP=(ISY2B-1)*4+JMJ                                              SPLI
      CALL SYMASI(1,K1,LIMK,L3B,M3B,JMK,NOBT,ISY3B)                     SPLI
      KEXP=(ISY3B-1)*4+JMK                                              SPLI
      IF(IND.EQ.0)RETURN                                                SPLI
      CALL IDNOM(I1,I2,J1,J2,IS1)                                       SPLI
      CALL IDNOM(K1,K2,1,1,IS2)                                         SPLI
      CALL SYMAS2(NOBT,6,IS3)                                           SPLI
      CALL SYMAS3(NOBT,6)                                               SPLI
      RETURN                                                            SPLI
      ENTRY SPLIT4(IND,NOBT,ISY1B,ISY2B,ISY3B,ISY4B)                    SPLI
      DO 3 JA=1,8                                                       SPLI
3     IDAR(JA,9)=IS3(JA)                                                SPLI
      CALL SYMASI(1,I1,LIMI,L1B,M1B,JMI,NOBT,ISY1B)                     SPLI
      IEXP=(ISY1B-1)*4+JMI                                              SPLI
      CALL SYMASI(1,J1,LIMJ,L2B,M2B,JMJ,NOBT,ISY2B)                     SPLI
      JEXP=(ISY2B-1)*4+JMJ                                              SPLI
      CALL SYMASI(1,K1,LIMK,L3B,M3B,JMK,NOBT,ISY3B)                     SPLI
      KEXP=(ISY3B-1)*4+JMK                                              SPLI
      CALL SYMASI(1,L1,LIML,L4B,M4B,JML,NOBT,ISY4B)                     SPLI
      LEXP=(ISY4B-1)*4+JML                                              SPLI
      IF(IND.EQ.0)RETURN                                                SPLI
      CALL IDNOM(I1,I2,J1,J2,IS1)                                       SPLI
      CALL IDNOM(K1,K2,L1,L2,IS2)                                       SPLI
      CALL SYMAS2(NOBT,8,IS3)                                           SPLI
      CALL SYMAS3(NOBT,8)                                               SPLI
      RETURN                                                            SPLI
      END                                                               SPLI
      SUBROUTINE SYMASI(INDEX,I,LIM,L,ML,JM1,NOBT,ISY)                  SYA1
      INTEGER NOBT(3)                                                   SYA1
      IF(INDEX.EQ.2)GOTO15                                              SYA1
      GO TO(10,11,10,10,10,12,11,11,11,10,10,10,10,10,13,12,12,12,11,11,SYA1
     .11,11,11),I                                                       SYA1
10    JM1=1                                                             SYA1
      GO TO 14                                                          SYA1
11    JM1=2                                                             SYA1
      GO TO 14                                                          SYA1
12    JM1=3                                                             SYA1
      GO TO 14                                                          SYA1
13    JM1=4                                                             SYA1
14    GO TO(1,1,2,3,4,1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9),I            SYA1
15     GOTO(1,2,3,4,5,6,7,8,9),I                                        SYA1
1     ISY=1                                                             SYA1
      LIM=NOBT(1)                                                       SYA1
      L=0                                                               SYA1
      ML=0                                                              SYA1
      RETURN                                                            SYA1
2     ISY=2                                                             SYA1
      LIM=NOBT(2)                                                       SYA1
      L=1                                                               SYA1
      ML=-1                                                             SYA1
      RETURN                                                            SYA1
3     ISY=2
```

```
      LIM=NOBT(2)                                                        SYA1
      L=1                                                                SYA1
      ML=0                                                               SYA1
      RETURN                                                             SYA1
4     ISY=2                                                              SYA1
      LIM=NOBT(2)                                                        SYA1
      L=1                                                                SYA1
      ML=1                                                               SYA1
      RETURN                                                             SYA1
5     ISY=3                                                              SYA1
      LIM=NOBT(3)                                                        SYA1
      L=2                                                                SYA1
      ML=-2                                                              SYA1
      RETURN                                                             SYA1
6     ISY=3                                                              SYA1
      LIM=NOBT(3)                                                        SYA1
      L=2                                                                SYA1
      ML=-1                                                              SYA1
      RETURN                                                             SYA1
7     ISY=3                                                              SYA1
      LIM=NOBT(3)                                                        SYA1
      L=2                                                                SYA1
      ML=0                                                               SYA1
      RETURN                                                             SYA1
8     ISY=3                                                              SYA1
      LIM=NOBT(3)                                                        SYA1
      L=2                                                                SYA1
      ML=1                                                               SYA1
      RETURN                                                             SYA1
9     ISY=3                                                              SYA1
      LIM=NOBT(3)                                                        SYA1
      L=2                                                                SYA1
      ML=2                                                               SYA1
      RETURN                                                             SYA1
      END                                                                SYA1
      FUNCTION SYMCHE(QN)                                                SYCH
      INTEGER SYMCHE,QN                                                  SYCH
      SYMCHE=2                                                           SYCH
      IF(QN.LE.2.OR.QN.EQ.6.OR.QN.EQ.15)SYMCHE=1                         SYCH
      IF(IABS(QN-12).LE.2.OR.IABS(QN-21).LE.2)SYMCHE=3                   SYCH
      RETURN                                                             SYCH
      END                                                                SYCH
      SUBROUTINE SYMAS2(NOBT,LDO6,IS3)                                   SYA2
      INTEGER IDX(4),NOBT(3),IM(2,6)/1,3,1,5,1,7,3,5,3,7,5,7/            SYA2
      INTEGER IS3(8),IS4(8)                                              SYA2
C IDAR CONTAINS IN ITS SECOND COLUMN THE SYMETRIES                       SYA2
C TO WHICH I2,I1,ETC. BELONG                                             SYA2
      COMMON/SYM/IDAR(8,10)                                              SYA2
      IJN(I,J)=I+(J*(J-1))/2                                             SYA2
      IJN1(I,J)=MIN0(I,J)+(MAX0(I,J)*(MAX0(I,J)-1))/2                    SYA2
      LDO1=LDO6-1                                                        SYA2
      LDO4=LDO6-3                                                        SYA2
C   A SYMMETRIC INDEX FOR EACH ELECTRON IS COMPUTED                      SYA2
      DO 1 JA=1,LDO6,2                                                   SYA2
      IDAR(JA,1)=JA                                                      SYA2
      IDAR(JA+1,1)=JA+1                                                  SYA2
1     IDAR(JA,3)=IJN1(IDAR(JA,2),IDAR(JA+1,2))                           SYA2
C                                                                        SYA2
C THE ID'S OF THE INTEGRALS ARE SORTED                                   SYA2
3     IX=0
```

```
      DO 4 JA=1,LDO4,2
      IF(IDAR(JA,3).LE.IDAR(JA+2,3))GOTO4                                  SYA2
      IX=1                                                                 SYA2
      DO 5 JB=1,3                                                          SYA2
      DO 5 JC=1,2                                                          SYA2
      IEX=IDAR(JA+JC-1,JB)                                                 SYA2
      IDAR(JA+JC-1,JB)=IDAR(JA+JC+1,JB)                                    SYA2
5     IDAR(JA+JC+1,JB)=IEX                                                 SYA2
      DO 11 JC=1,2                                                         SYA2
      IEX=IDAR(JA+JC-1,9)                                                  SYA2
      IDAR(JA+JC-1,9)=IDAR(JA+JC+1,9)                                      SYA2
11    IDAR(JA+JC+1,9)=IEX                                                  SYA2
4     CONTINUE                                                             SYA2
      IF(IX.EQ.1)GOTO3                                                     SYA2
C   THE ELECTRONS ARE RESORTED S.T. OF TWO ELECTRONS WITH EQUAL I.D.       SYA2
C   IN IDAR(*,3)THE ONE WITH THE SMALLER STARTING# IS FIRST                SYA2
30    IX=0                                                                 SYA2
      DO 2 JA=1,LDO4,2                                                     SYA2
      IF(IDAR(JA,3).NE.IDAR(JA+2,3))GOTO2                                  SYA2
      IF(IDAR(JA,2).LE.IDAR(JA+2,2))GOTO2                                  SYA2
      IX=1                                                                 SYA2
      DO 31 JB=1,3                                                         SYA2
      DO 31 JC=1,2                                                         SYA2
      IEX=IDAR(JA+JC-1,JB)                                                 SYA2
      IDAR(JA+JC-1,JB)=IDAR(JA+JC+1,JB)                                    SYA2
31    IDAR(JA+JC+1,JB)=IEX                                                 SYA2
      DO 32 JC=1,2                                                         SYA2
      IEX=IDAR(JA+JC-1,9)                                                  SYA2
      IDAR(JA+JC-1,9)=IDAR(JA+JC+1,9)                                      SYA2
32    IDAR(JA+JC+1,9)=IEX                                                  SYA2
2     CONTINUE                                                             SYA2
      IF(IX.EQ.1)GOTO30                                                    SYA2
C   AN ID FOR THE ORBITALS IS COMPUTED                                     SYA2
C   1S=1 2S=2...2P=5 3P=6...3D=8 4D=9                                      SYA2
120   DO 121 JA=1,LDO6                                                     SYA2
      DO 121 JB=1,2                                                        SYA2
      IF(JB.EQ.1)ISIG=IDAR(JA,9)                                           SYA2
      IF(JB.EQ.2)ISIG=IS3(JA)                                              SYA2
      GO TO(13,14,17,17,17,15,18,18,18,20,20,20,20,20,16,19,19,19,21,21,SYA2
     .21,21,21),ISIG                                                       SYA2
13    IDA=1                                                                SYA2
      GOTO12                                                               SYA2
14    IDA=2                                                                SYA2
      GOTO12                                                               SYA2
15    IDA=3                                                                SYA2
      GOTO12                                                               SYA2
16    IDA=4                                                                SYA2
      GOTO12                                                               SYA2
17    IDA=5                                                                SYA2
      GOTO12                                                               SYA2
18    IDA=6                                                                SYA2
      GOTO12                                                               SYA2
19    IDA=7                                                                SYA2
      GOTO12                                                               SYA2
20    IDA=8                                                                SYA2
      GOTO12                                                               SYA2
21    IDA=9                                                                SYA2
12    IF(JB.EQ.1)IDAR(JA,10)=IDA                                           SYA2
121   IF(JB.EQ.2)IS4(JA)=IDA                                               SYA2
C   THE PARTNER FOR EACH BRA IS FOUND
```

```
      IY=9
200   DO 210 JA=1,LD06,2                                             SYA2
      IF(IS3(JA).NE.IS3(JA+1))GOTO210                                SYA2
      DO 211 IA=1,LD06,2                                             SYA2
      IF(IS3(JA).NE.IDAR(IA,IY))GOTO211                              SYA2
      IF(IS3(JA+1).NE.IDAR(IA+1,IY))GOTO211                          SYA2
      IDAR(IA,IY)=0                                                  SYA2
      IDAR(IA+1,IY)=0                                                SYA2
      IDAR(IA,1)=JA                                                  SYA2
      IDAR(IA+1,1)=JA+1                                              SYA2
      GOTO210                                                        SYA2
211   CONTINUE                                                       SYA2
210   CONTINUE                                                       SYA2
C  LOOP 220 CHECKS IF THE INTEGRAL IS OF THE EXCHANGED KIND          SYA2
      ISUM=0                                                         SYA2
      DO 220 JA=1,LD06                                               SYA2
220   ISUM=ISUM+IDAR(JA,IY)                                          SYA2
      IF(ISUM.EQ.0)RETURN                                            SYA2
C   THE LOOPS 230 HANDLE TWO-CYCLE EXCHANGE                          SYA2
      DO 230 JA=1,LD06,2                                             SYA2
      IF(IS3(JA).EQ.IS3(JA+1))GOTO230                                SYA2
      DO 231 IA=1,LD06,2                                             SYA2
      IF(IS3(JA).NE.IDAR(IA,IY))GOTO231                              SYA2
      IF(IS3(JA+1).NE.IDAR(IA+1,IY))GOTO231                          SYA2
      DO 232 IB=1,LD06,2                                             SYA2
      IF(IS3(JA).NE.IDAR(IB+1,IY))GOTO232                            SYA2
      IF(IS3(JA+1).NE.IDAR(IB,IY))GOTO232                            SYA2
      JLIM=JA+2                                                      SYA2
      IF(JLIM.GT.LD06)GOTO230                                        SYA2
      DO 233 JB=JLIM,LD06,2                                          SYA2
      IF(IS3(JB).NE.IS3(JA+1))GOTO233                                SYA2
      IF(IS3(JB+1).NE.IS3(JA))GOTO233                                SYA2
      IDAR(IA,IY)=0                                                  SYA2
      IDAR(IA+1,IY)=0                                                SYA2
      IDAR(IB,IY)=0                                                  SYA2
      IDAR(IB+1,IY)=0                                                SYA2
      IDAR(IA,1)=JA                                                  SYA2
      IDAR(IA+1,1)=JB+1                                              SYA2
      IDAR(IB,1)=JB                                                  SYA2
      IDAR(IB+1,1)=JA+1                                              SYA2
      GOTO230                                                        SYA2
233   CONTINUE                                                       SYA2
      GOTO230                                                        SYA2
232   CONTINUE                                                       SYA2
      GOTO230                                                        SYA2
231   CONTINUE                                                       SYA2
230   CONTINUE                                                       SYA2
C LOOP 240 CHECKS IF THE INTEGRAL CONTAINS TRIPLE OR QUADRUPLE EXCHANGE SYA2
      ISUM=0                                                         SYA2
      DO 240 JA=1,LD06                                               SYA2
240   ISUM=ISUM+IDAR(JA,IY)                                          SYA2
      IF(ISUM.EQ.0)RETURN                                            SYA2
C  LOOPS 250 HANDLE THE THREE&FOUR-CYCLE EXCHANGE                    SYA2
      DO 250 JA=1,LD06,2                                             SYA2
      IF(IS3(JA).EQ.IS3(JA+1))GOTO250                                SYA2
      IRE=0                                                          SYA2
252   DO 251 IA=1,LD06,2                                             SYA2
      IF(IS3(JA).NE.IDAR(IA+IRE,IY))GOTO251                          SYA2
      IDAR(IA+IRE,IY)=0                                              SYA2
      IDAR(IA+IRE,1)=JA+IRE
```

```
      IRE=IRE+1                                                         SYA2
      IF(IRE.EQ.2)GOTO250                                               SYA2
      GOTO252                                                           SYA2
251   CONTINUE                                                          SYA2
250   CONTINUE                                                          SYA2
C   LOOP 260 CHECKS IF ALL ORBITAL HAVE AGREED IN THREE QN'S            SYA2
      ISUM=0                                                            SYA2
      DO 260 JA=1,LDO6                                                  SYA2
260   ISUM=ISUM+IDAR(JA,IY)                                             SYA2
      IF(ISUM.EQ.0)RETURN                                               SYA2
      IF(IY.EQ.10)GOTO261                                               SYA2
      IY=10                                                             SYA2
      DO 262 JA=1,LDO6                                                  SYA2
262   IS3(JA)=IS4(JA)                                                   SYA2
      GOTO200                                                           SYA2
261   WRITE(6,900)(IS3(JA),JA=1,LDO6)                                   SYA2
      WRITE(6,901)((IDAR(JA,JB),JB=1,10),JA=1,LDO6)                     SYA2
      STOP                                                              SYA2
900   FORMAT(65(' *')/40X,'ERROR IN INTEGRALSORTING'/65(' *')//4(3X,2I3)SYA2
     .)                                                                 SYA2
901   FORMAT(' ',10I4)                                                  SYA2
      ENTRY SYMAS3(NOBT,LDO6)                                           SYA2
C   THE VALUES LIM1B,L1B,ETC. ARE COMPUTED                              SYA2
      DO 6 JA=1,LDO6                                                    SYA2
6     CALL SYMASI(2,IDAR(JA,2),IDAR(JA,4),IDAR(JA,5),IDAR(JA,6),JM,NOBT,SYA2
     .IDAR(JA,7))                                                       SYA2
      IF(LDO6-6)8,9,10                                                  SYA2
10    DO 7 JA=1,6                                                       SYA2
7     IDAR(JA,8)=IJN1(IDAR(IM(1,JA),3),IDAR(IM(2,JA),3))                SYA2
      RETURN                                                            SYA2
8     IDAR(1,8)=IJN(IDAR(1,3),IDAR(3,3))                                SYA2
      RETURN                                                            SYA2
9     IDAR(2,3)=IJN1(IDAR(3,3),IDAR(5,3))                               SYA2
      IDAR(4,3)=IJN1(IDAR(1,3),IDAR(5,3))                               SYA2
      IDAR(6,3)=IJN1(IDAR(1,3),IDAR(3,3))                               SYA2
      RETURN                                                            SYA2
      END                                                               SYA2
      SUBROUTINE IDNOM(IA,IB,JA,JB,IS)                                  IDNO
      INTEGER IS(4),SYMCHE,IV(4)                                        IDNO
      IV(1)=IA                                                          IDNO
      IV(2)=IB                                                          IDNO
      IV(3)=JA                                                          IDNO
      IV(4)=JB                                                          IDNO
      DO 1 J1=1,4                                                       IDNO
      IF(SYMCHE(IV(J1))-2)2,3,4                                         IDNO
2     IS(J1)=1                                                          IDNO
      GOTO1                                                             IDNO
3     IS(J1)=IV(J1)-4*(IV(J1)/7)+(IV(J1)/16)-1                          IDNO
      GOTO1                                                             IDNO
4     IS(J1)=IV(J1)-5*(IV(J1)/9)-4*(IV(J1)/19)                          IDNO
1     CONTINUE                                                          IDNO
      RETURN                                                            IDNO
      END                                                               SYA3
      SUBROUTINE SYM34(NOBT,LOI,LOJ,LOK,LOL,ISYM,*,LIMIT)               SYA3
      INTEGER ISV(4),NOBT(3),IV(8),IVM(4)                               SYA3
      LOGICAL LC(4),LOI,LOJ,LOK,LOL,LOG                                 SYA3
      COMMON/SYM/IDAR(8,10)                                             SYA3
      COMMON/SPLI1/I1,I2,J1,J2,K1,K2,L1,L2,LIMI,LIMJ,LIMK,LIML,JMI,JMJ,JSYA3
     .MK,JML,IEXP,JEXP,KEXP,LEXP                                        SYA3
      EQUIVALENCE(IV(1),I1),(IVM(1),JMI)
```

```
      LOG=.FALSE.
      IF(LIMIT.EQ.4)CALL SPLIT4(0,NOBT,ISV(1),ISV(2),ISV(3),ISV(4))     SYA3
      IF(LIMIT.EQ.3)CALL SPLIT3(0,NOBT,ISV(1),ISV(2),ISV(3))            SYA3
      DO 1 JA=1,LIMIT                                                   SYA3
      LC(JA)=ISV(JA).EQ.ISYM                                            SYA3
1     LOG=LOG.OR.LC(JA)                                                 SYA3
      IF(.NOT.LOG)RETURN1                                               SYA3
11    IX=0                                                              SYA3
      DO 2 JA=2,LIMIT                                                   SYA3
      IF(LC(JA))GOTO2                                                   SYA3
      IF(.NOT.LC(JA-1))GOTO2                                            SYA3
      IX=1                                                              SYA3
      LOG=LC(JA-1)                                                      SYA3
      LC(JA-1)=LC(JA)                                                   SYA3
      LC(JA)=LOG                                                        SYA3
      IEX=IV(JA+JA-2)                                                   SYA3
      IV(JA+JA-2)=IV(JA+JA)                                             SYA3
      IV(JA+JA)=IEX                                                     SYA3
      IEX=IV(JA+JA-3)                                                   SYA3
      IV(JA+JA-3)=IV(JA+JA-1)                                           SYA3
      IV(JA+JA-1)=IEX                                                   SYA3
2     CONTINUE                                                          SYA3
      IF(IX.EQ.1)GOTO11                                                 SYA3
      IF(LIMIT.EQ.4)CALL SPLIT4(0,NOBT,ISV(1),ISV(2),ISV(3),ISV(4))     SYA3
      IF(LIMIT.EQ.3)CALL SPLIT3(0,NOBT,ISV(1),ISV(2),ISV(3))            SYA3
      DO 4 JA=1,LIMIT                                                   SYA3
4     LC(JA)=ISV(JA).EQ.ISYM                                            SYA3
      LOG=.FALSE.                                                       SYA3
      DO 8 JA=2,LIMIT                                                   SYA3
8     LOG=LOG.OR.LC(JA-1)                                               SYA3
      IF(.NOT.LOG)GOTO7                                                 SYA3
      LUP=LIMIT-1                                                       SYA3
      IS=LUP                                                            SYA3
      DO 9 JA=1,LUP                                                     SYA3
9     IF(LC(LIMIT-JA))IS=LIMIT-JA                                       SYA3
      IF(LC(1))IS=1                                                     SYA3
5     IX=0                                                              SYA3
      DO 6 JA=IS,LUP                                                    SYA3
      IF(IVM(JA).LE.IVM(JA+1))GOTO6                                     SYA3
      IX=1                                                              SYA3
      IEX=IVM(JA)                                                       SYA3
      IVM(JA)=IVM(JA+1)                                                 SYA3
      IVM(JA+1)=IEX                                                     SYA3
      IEX=IV(JA+JA-1)                                                   SYA3
      IV(JA+JA-1)=IV(JA+JA+1)                                           SYA3
      IV(JA+JA+1)=IEX                                                   SYA3
      IEX=IV(JA+JA)                                                     SYA3
      IV(JA+JA)=IV(JA+JA+2)                                             SYA3
      IV(JA+JA+2)=IEX                                                   SYA3
6     CONTINUE                                                          SYA3
      IF(IX.EQ.1)GOTO5                                                  SYA3
7     IF(LIMIT.EQ.3)GOTO10                                              SYA3
      CALL SPLIT4(1,NOBT,IS1,IS2,IS3,IS4)                               SYA3
      LOI=IS1.EQ.ISYM                                                   SYA3
      LOJ=IS2.EQ.ISYM                                                   SYA3
      LOK=IS3.EQ.ISYM                                                   SYA3
      LOL=IS4.EQ.ISYM                                                   SYA3
      RETURN                                                            SYA3
10    CALL SPLIT3(1,NOBT,IS1,IS2,IS3)                                   SYA3
      LOI=IS1.EQ.ISYM
```

```
      LOJ=IS2.EQ.ISYM                                                   SYA3
      LOK=IS3.EQ.ISYM                                                   SYA3
      RETURN                                                            SYA3
      END                                                               SYA3
C  USED FOR DIRECT-ACCESS, PROGRAMMED BY L.THIEL,COMP.CENT. U OF A      LIOU
         MACRO                                                          LIOU
         SETUP &EPN                                                     LIOU
         USING SAVE,13                                                  LIOU
         B     14(15)                                                   LIOU
         DC    X'08'                                                    LIOU
         DC    CL8&EPN                                                  LIOU
         STM   14,12,12(13)                                             LIOU
         LR    11,13                                                    LIOU
         LA    13,36(15)                                                LIOU
         ST    11,SAVE+4                                                LIOU
         ST    13,8(11)                                                 LIOU
         B     STWORK                                                   LIOU
SAVE     DC    18F'0'                                                   LIOU
         MEND                                                           LIOU
LOGIOU   CSECT                                                          LIOU
*        CALL LOGIOU(INFO,LOGU,&RTN)                                    LIOU
*        GENERATE A CALL TO GDINFO FOR A LOGICAL I/O UNIT AND RETURN    LIOU
*        THE INFORMATION SUPPLIED TO THE USER.                          LIOU
*        INFO - A 16 BYTE REGION AS DESCRIBED IN "MTS SYSTEM            LIOU
*              SUBROUTINES" UNDER GDINFO(NORMAL RETURN).                LIOU
*        LOGU - ANY MTS LOGICAL I/O UNIT, LEFT JUSTIFIED WITH           LIOU
*              TRAILING BLANKS.                                         LIOU
*        ALTERNATE RETURN - NAME IS NOT A LEGAL LOGICAL I/O UNIT OR     LIOU
*              NO DEVICE WAS ASSIGNED TO THE UNIT.                      LIOU
*              NO INFORMATION RETURNED IN THIS CASE.                    LIOU
         SETUP 'LOGIOU'                                                 LIOU
VGD      DC    V(GDINFO)                                                LIOU
VFREE    DC    V(FREESPAC)                                              LIOU
STWORK   LM    2,3,0(1)          PARM LIST                              LIOU
         L     15,VGD            ADDR OF GDINFO                         LIOU
         LM    0,1,0(3)          LOGICAL I/O UNIT                       LIOU
         BALR  14,15                                                    LIOU
         LTR   10,15             TEST                                   LIOU
         BNZ   RTN                    RC                                LIOU
         MVC   0(16,2),0(1)      MOVE INFO                              LIOU
         SR    0,0               FREE                                   LIOU
         L     15,VFREE               INFO                              LIOU
         BALR  14,15                  REGION                            LIOU
         LR    15,10                                                    LIOU
RTN      L     13,SAVE+4         GDINFO RC                              LIOU
         LR    15,10                                                    LIOU
         L     14,12(13)         RETURN ADDR                            LIOU
         LM    0,12,20(13)       RELOAD R0 - R12                        LIOU
         BR    14                                                       LIOU
         END                                                            LIOU
C THE FOLLOWING ROUTINES COMPUTE INTEGRALS OVER SLATER-TYPE-ORBITALS    ONIN
C THEY HAVE BEEN CODED BY F.W.BIRSS, DEPT.CHEM., U OF A                 ONIN
      SUBROUTINE ONEI(NB,LB,MB,CB,NK,LK,MK,CK,Z,S,H,HH,RM1,RP1,RP2)     ONIN
C     ONE ELECTRON INTEGRALS                                            ONIN
C     CFACT ASSUMED                                                     ONIN
      IMPLICIT REAL*8(A-H,O-Z)                                          ONIN
      REAL*8 FC(10)/4.D0,1.333333333333333,2.666666666666666,0.8D0,4.8D0ONIN
     .,19.2D0,.5714285714285714,6.857142857142858,68.57142857142858,411.ONIN
     .4285714285714/                                                    ONIN
      INTEGER ID/0/                                                     ONIN
```

```
      COMMON/CFACT/FACT(41)                                            ONIN
      IF(ID.EQ.1)GOTO2                                                 ONIN
      ID=1                                                             ONIN
      FACT(1)=1.D0                                                     ONIN
      DO 1 I=2,41                                                      ONIN
    1 FACT(I)=DFLOAT(I-1)*FACT(I-1)                                    ONIN
    2 S=0.0D0                                                          ONIN
      H=0.0D0                                                          ONIN
      HH=0.0D0                                                         ONIN
      RM1=0.D0                                                         ONIN
      RP1=0.D0                                                         ONIN
      RP2=0.D0                                                         ONIN
      IF(LB.NE.LK.OR.MB.NE.MK.OR.IABS(MB).GT.LB)RETURN                 ONIN
      C=CB+CK                                                          ONIN
      N=NB+NK+1                                                        ONIN
      CP=C**N                                                          ONIN
      CPM1=C**(N-1)                                                    ONIN
      CPP1=C**(N+1)                                                    ONIN
      CPP2=C**(N+2)                                                    ONIN
      L=LB*(LB+1)                                                      ONIN
      ENB=(L-NB*(NB-1))*0.5D0                                          ONIN
      ENK=(L-NK*(NK-1))*0.5D0                                          ONIN
      CZB=CB*NB-Z                                                      ONIN
      CZK=CK*NK-Z                                                      ONIN
      CSB=CB*CB*0.5D0                                                  ONIN
      CSK=CK*CK*0.5D0                                                  ONIN
      FT=FC((LB*(LB+1))/2+IABS(MB)+1)                                  ONIN
      S=FT*FACT(N)/CP                                                  ONIN
      RM1=FT*FACT(N-1)/CPM1                                            ONIN
      RP1=FT*FACT(N+1)/CPP1                                            ONIN
      RP2=FT*FACT(N+2)/CPP2                                            ONIN
      H=FT*((ENK*FACT(N-2)*C+CZK*FACT(N-1))*C-CSK*FACT(N))/CP          ONIN
      IF(N.GT.4)HH=ENB*ENK*FACT(N-4)*C                                 ONIN
      IF(N.GT.3)HH=(HH+(ENB*CZK+ENK*CZB)*FACT(N-3))*C                  ONIN
      HH=FT*(((HH+(CZB*CZK-ENB*CSK-ENK*CSB)*FACT(N-2))*C-(CZB*CSK+CZK*CSONIN
     .B)*FACT(N-1))*C+CSB*CSK*FACT(N))/CP                              ONIN
      RETURN                                                           ONIN
      END                                                              ONIN
      SUBROUTINE HR(N1B,L1B,M1B,C1B,N2B,L2B,M2B,C2B,N1K,L1K,M1K,C1K,N2K,HRIN
     .L2K,M2K,C2K,Z,T1,T2)                                             HRIN
C     MIXED H AND 1/R(1,2) INTEGRALS                                   HRIN
C     REPI REQUIRED                                                    HRIN
      IMPLICIT REAL*8(A-H,O-Z)                                         HRIN
      EN1=0.5D0*(L1B*(L1B+1)-N1B*(N1B-1)+L1K*(L1K+1)-N1K*(N1K-1))      HRIN
      EN2=0.5D0*(L2B*(L2B+1)-N2B*(N2B-1)+L2K*(L2K+1)-N2K*(N2K-1))      HRIN
      CZ1=C1B*N1B+C1K*N1K-Z-Z                                          HRIN
      CZ2=C2B*N2B+C2K*N2K-Z-Z                                          HRIN
      CS1=0.5D0*(C1B*C1B+C1K*C1K)                                      HRIN
      CS2=0.5D0*(C2B*C2B+C2K*C2K)                                      HRIN
      A=0.0D0                                                          HRIN
      IF(EN1.NE.0.0D0)A=REPI(1,N1B-2,L1B,M1B,C1B,N2B,L2B,M2B,C2B,N1K,L1KHRIN
     .,M1K,C1K,N2K,L2K,M2K,C2K,1,0,0,1.D0,1,0,0,1.D0)                  HRIN
      B=REPI(1,N1B-1,L1B,M1B,C1B,N2B,L2B,M2B,C2B,N1K,L1K,M1K,C1K,N2K,L2KHRIN
     .,M2K,C2K,1,0,0,1.D0,1,0,0,1.D0)                                  HRIN
      C=REPI(1,N1B,L1B,M1B,C1B,N2B,L2B,M2B,C2B,N1K,L1K,M1K,C1K,N2K,L2K,MHRIN
     .2K,C2K,1,0,0,1.D0,1,0,0,1.D0)                                    HRIN
      F=0.0D0                                                          HRIN
      IF(EN2.NE.0.0D0)F=REPI(1,N1B,L1B,M1B,C1B,N2B-2,L2B,M2B,C2B,N1K,L1KHRIN
     .,M1K,C1K,N2K,L2K,M2K,C2K,1,0,0,1.D0,1,0,0,1.D0)                  HRIN
      G=REPI(1,N1B,L1B,M1B,C1B,N2B,L2B,M2B,C2B,N1K,L1K,M1K,C1K,N2K-1,L2KHRIN
```

```
     .,M2K,C2K,1,0,0,1.D0,1,0,0,1.D0)                                   HRIN
      T1=EN1*A+CZ1*B-CS1*C                                              HRIN
      T2=EN2*F+CZ2*G-CS2*C                                              HRIN
      RETURN                                                            HRIN
      END                                                               HRIN
      FUNCTION REPI(IND,NLA,LLA,MLA,CLA,NRA,LRA,MRA,CRA,NLB,LLB,MLB,CLB,REPI
     1NRB,LRB,MRB,CRB,NCA,LCA,MCA,CCA,NCB,LCB,MCB,CCB)                  REPI
C     ONE-CENTRE TWO- AND THREE-ELECTRON INTEGRAL FUNCTION (IMAGINARY)  REPI
C     ANGLI REQUIRED                                                    REPI
      IMPLICIT REAL*8 (A-H,O-Z)                                         REPI
      REAL*8 FC(325,5),PL(9,5),ST(55),TP(45)                            REPI
      INTEGER*4 IU/0/,LT(3),MT(3)                                       REPI
      INTEGER*2 IC(325,5)/1575*0/                                       REPI
      COMMON /CFACT/ FACT(41)                                           REPI
      COMMON /CPSI/ PSI(11)                                             REPI
      IF (IU.EQ.1) GO TO 7                                              REPI
      IU=1                                                              REPI
      FACT(1)=1.D0                                                      REPI
      DO 1 I=1,40                                                       REPI
    1 FACT(I+1)=I*FACT(I)                                               REPI
      PSI(1)=0.0D0                                                      REPI
      DO 2 I=1,10                                                       REPI
    2 PSI(I+1)=PSI(I)+1.D0/I                                            REPI
      W=1.D0                                                            REPI
      DO 4 LP=1,9                                                       REPI
      W=0.5D0*W                                                         REPI
      MA=(LP+1)/2                                                       REPI
      ML=LP+LP-1                                                        REPI
      Y=(-1.D0)**MA*ML*W                                                REPI
      DO 3 MP=1,MA                                                      REPI
      Y=-Y                                                              REPI
      MB=MA-MP                                                          REPI
      MC=LP-MB                                                          REPI
      MD=MC-MB                                                          REPI
    3 PL(LP,MP)=Y*FACT(MD+LP-1)/(FACT(MB+1)*FACT(MC)*FACT(MD))          REPI
      DO 4 MP=1,LP                                                      REPI
    4 TP((LP*(LP+1))/2-MP+1)=16.D0*FACT(LP+MP-1)/(FACT(LP-MP+1)*ML**2)  REPI
      DO 6 LXP=1,5                                                      REPI
      LT(1)=LXP-1                                                       REPI
      MA=LXP+LXP-1                                                      REPI
      DO 6 MXP=1,MA                                                     REPI
      MX=MXP-LXP                                                        REPI
      MT(1)=IABS(MX)                                                    REPI
      LMX=LXP*(LXP-1)+1-MX                                              REPI
      DO 6 LYP=LXP,5                                                    REPI
      LT(2)=LYP-1                                                       REPI
      MC=LYP+LYP-1                                                      REPI
      LMAXP=LXP+LYP                                                     REPI
      DO 6 MYP=1,MC                                                     REPI
      MY=MYP-LYP                                                        REPI
      MT(2)=IABS(MY)                                                    REPI
      LMY=LYP*(LYP-1)+1-MY                                              REPI
      IF (LMX.GT.LMY) GO TO 6                                           REPI
      LMXY=(LMY*(LMY-1))/2+LMX                                          REPI
      IPLC=0                                                            REPI
      DO 5 LSP=2,LMAXP,2                                                REPI
      LB=LMAXP-LSP                                                      REPI
      LT(3)=LB                                                          REPI
      MB=MY-MX                                                          REPI
      MBA=IABS(MB)
```

```
      IF (MBA.GT.LB) GO TO 5                                            REPI
      MT(3)=MBA                                                         REPI
      X=ANGLI(LT,MT)                                                    REPI
      IF(X.EQ.0.0D0) GO TO 5                                            REPI
      IPLC=IPLC+1                                                       REPI
      IC(LMXY,IPLC)=LB*(LB+1)+1-MBA                                     REPI
      FC(LMXY,IPLC)=X*(LB+LB+1)*FACT(LB-MBA+1)/FACT(LB+MBA+1)           REPI
    5 CONTINUE                                                          REPI
    6 CONTINUE                                                          REPI
    7 REPI=0.0D0                                                        REPI
      IF (IND.LT.3.AND.MLA+MRA.NE.MLB+MRB) RETURN                       REPI
      MA=LLA*(LLA+1)+1-MLA                                              REPI
      MB=LLB*(LLB+1)+1-MLB                                              REPI
      IL=MAX0(MA,MB)                                                    REPI
      IL=(IL*(IL-3))/2+MA+MB                                            REPI
      NL=NLA+NLB                                                        REPI
      CL=CLA+CLB                                                        REPI
      MA=LRA*(LRA+1)+1-MRA                                              REPI
      MB=LRB*(LRB+1)+1-MRB                                              REPI
      IR=MAX0(MA,MB)                                                    REPI
      IR=(IR*(IR-3))/2+MA+MB                                            REPI
      NR=NRA+NRB                                                        REPI
      CR=CRA+CRB                                                        REPI
      GO TO (8,9,34),IND                                                REPI
    8 C=1.D0/(CL+CR)                                                    REPI
      CA=C*CL                                                           REPI
      CB=C*CR                                                           REPI
      C=C**(NL+NR+1)                                                    REPI
      GO TO 11                                                          REPI
    9 IF(CL.LE.CR) GO TO 10                                             REPI
      MA=NL                                                             REPI
      NL=NR                                                             REPI
      NR=MA                                                             REPI
      X=CL                                                              REPI
      CL=CR                                                             REPI
      CR=X                                                              REPI
   10 C=CR/CL                                                           REPI
      CS=C*C                                                            REPI
      V=-DLOG(C)                                                        REPI
      CA=C/(C+1.D0)                                                     REPI
      CB=C-1.D0                                                         REPI
      IF (CB.GT.0.0D0) CB=C/CB                                          REPI
   11 KP=-1                                                             REPI
      DO 33 I=1,5                                                       REPI
      MA=IC(IL,I)                                                       REPI
      IF (MA.LT.1) RETURN                                               REPI
      DO 32 J=1,5                                                       REPI
      IF (MA-IC(IR,J))32,12,33                                          REPI
   12 K=SQRT(FLOAT(MA-1)+0.001)                                         REPI
      MU=((K+1)*(K+2))/2-IABS(K*(K+1)+1-MA)                             REPI
      IF (K.EQ.KP) GO TO 31                                             REPI
      IF (IND.EQ.2) GO TO 15                                            REPI
      MA=NR-K                                                           REPI
      MC=NL+K                                                           REPI
      CR=1.D0                                                           REPI
      SA=0.0D0                                                          REPI
      DO 13 L=1,MA                                                      REPI
      CR=CR*CB                                                          REPI
   13 SA=SA+FACT(MC+L)*CR/FACT(L)                                       REPI
      MB=NL-K
```

```
      MC=NR+K
      CL=1.D0                                                          REPI
      SB=0.0D0                                                         REPI
      DO 14 L=1,MB                                                     REPI
      CL=CL*CA                                                         REPI
   14 SB=SB+FACT(MC+L)*CL/FACT(L)                                      REPI
      SA=C*(SA*FACT(MA)/(CB*CR)+SB*FACT(MB)/(CA*CL))                   REPI
      GO TO 30                                                         REPI
   15 MB=K+1                                                           REPI
      DO 27 LP=1,MB                                                    REPI
      IA=LP+LP-3                                                       REPI
      DO 27 MP=LP,MB                                                   REPI
      LM=(MP*(MP-1))/2+LP                                              REPI
      KM=K-2*(MP-LP)                                                   REPI
      IBP=NL-KM                                                        REPI
      IB=IBP-1                                                         REPI
      SA=0.0D0                                                         REPI
      IF (IA.GT.0) GO TO 25                                            REPI
      DO 24 II=1,IBP                                                   REPI
      SB=0.0D0                                                         REPI
      Y=1.D0/C                                                         REPI
      Z=1.D0                                                           REPI
      IIM=IBP-II+1                                                     REPI
      DO 23 LL=1,IIM                                                   REPI
      Y=Y*C                                                            REPI
      MD=NR+KM+LL-1                                                    REPI
      SC=0.0D0                                                         REPI
      CAM=1.D0                                                         REPI
      Z=-Z                                                             REPI
      IF (II.GT.1) GO TO 21                                            REPI
      CBM=Z                                                            REPI
      IF(CB.GT.4.D0) CBM=0.0D0                                         REPI
      DO 16 NP=1,MD                                                    REPI
      X=V                                                              REPI
      IF (MD.NE.NP) X=1.D0/(MD-NP)                                     REPI
      CAM=CAM*CA                                                       REPI
      CBM=CBM*CB                                                       REPI
   16 SC=SC+X*(CAM+CBM)                                                REPI
      IF (C.NE.1.D0) GO TO 17                                          REPI
      SC=SC-Z/MD                                                       REPI
      GO TO 20                                                         REPI
   17 IF (CB.LE.4.D0) GO TO 20                                         REPI
      CBM=CB                                                           REPI
      CAM=1.D0/CB                                                      REPI
      SD=0.0D0                                                         REPI
      DO 18 NP=1,10000                                                 REPI
      CBM=CBM*CAM                                                      REPI
      X=CBM/(MD+NP-1)                                                  REPI
      SD=SD+X                                                          REPI
      IF (X/SD.LT.1.D-18) GO TO 19                                     REPI
   18 CONTINUE                                                         REPI
   19 SC=SC-Z*SD                                                       REPI
   20 SC=FACT(MD)*SC                                                   REPI
      GO TO 23                                                         REPI
   21 X=0.5D0*Z/C                                                      REPI
      U=1.D0                                                           REPI
      SC=0.0D0                                                         REPI
      IM=II-1                                                          REPI
      DO 22 NP=1,IM                                                    REPI
      X=X*C
```

```
      U=-U                                                              REPI
   22 IF (U.NE.Z) SC=SC-FACT(MD+NP-1)/(X*FACT(NP))                      REPI
   23 SB=SB+SC/(Y*FACT(LL))                                             REPI
   24 SA=SA+SB*FACT(IBP)/DMAX1(1.D0,DFLOAT(II-1))                       REPI
      GO TO 27                                                          REPI
   25 IAP=IA+1                                                          REPI
      ME=NR+KM+1                                                        REPI
      CBM=2.D0*FACT(IAP)/C                                              REPI
      DO 26 II=2,IAP,2                                                  REPI
      CBM=CBM*CS                                                        REPI
   26 SA=SA+FACT(IB+II)*FACT(ME-II)*CBM/(FACT(IAP-II+2)*FACT(II))       REPI
   27 ST(LM)=SA/(CL**NL*CR**NR*C**KM)                                   REPI
      MC=(1+(-1)**K)/2                                                  REPI
      MD=K/2+1                                                          REPI
      SA=PL(MB,1)*ST(MD)*MC                                             REPI
      MD=MD-MC                                                          REPI
      ME=MC                                                             REPI
      DO 29 L=1,K                                                       REPI
      LK=MB-L                                                           REPI
      DO 28 M=1,LK                                                      REPI
      DO 28 N=M,LK                                                      REPI
      MN=(N*(N-1))/2+M                                                  REPI
   28 ST(MN)=0.5D0*(ST(MN)+ST(MN+N)-ST(MN+N+1))                         REPI
      IF (MC.NE.0) GO TO 29                                             REPI
      ME=ME+1                                                           REPI
      SA=SA+PL(MB,ME)*ST(MD)                                            REPI
      MD=MD-1                                                           REPI
   29 MC=1-MC                                                           REPI
   30 KP=K                                                              REPI
   31 REPI=REPI+TP(MU)*FC(IL,I)*FC(IR,J)*SA                             REPI
   32 CONTINUE                                                          REPI
   33 CONTINUE                                                          REPI
      RETURN                                                            REPI
   34 IF (MCB-MCA.NE.MLA-MLB+MRA-MRB) RETURN                            REPI
      CC=CCA+CCB                                                        REPI
      S=CC+CL+CR                                                        REPI
      WA=S/CL                                                           REPI
      WB=S/CR                                                           REPI
      RA=CC+CL                                                          REPI
      RB=CC+CR                                                          REPI
      AAM=RA/CL                                                         REPI
      ABM=RA/CR                                                         REPI
      BAM=RB/CL                                                         REPI
      BBM=RB/CR                                                         REPI
      NC=NCA+NCB                                                        REPI
      NCP=NC+NL+NR+1                                                    REPI
      MA=LCA*(LCA+1)+1-MCA                                              REPI
      MB=LCB*(LCB+1)+1-MCB                                              REPI
      IM=MAX0(MA,MB)                                                    REPI
      IM=(IM*(IM-3))/2+MA+MB                                            REPI
      LSA=-1                                                            REPI
      LSB=-1                                                            REPI
      DO 40 I=1,5                                                       REPI
      MA=IC(IL,I)                                                       REPI
      IF(MA.LT.1) RETURN                                                REPI
      LL=SQRT(FLOAT(MA-1)+0.001)                                        REPI
      LT(1)=LL                                                          REPI
      MT(1)=IABS(LL*(LL+1)+1-MA)                                        REPI
      DO 39 J=1,5                                                       REPI
      MA=IC(IR,J)
```

```
      IF(MA.LT.1) GO TO 40
      LR=SQRT(FLOAT(MA-1)+0.001)                                       REPI
      LT(2)=LR                                                         REPI
      MT(2)=IABS(LR*(LR+1)+1-MA)                                       REPI
      DO 38 K=1,5                                                      REPI
      MA=IC(IM,K)                                                      REPI
      IF(MA.LT.1) GO TO 39                                             REPI
      LM=SQRT(FLOAT(MA-1)+0.001)                                       REPI
      LT(3)=LM                                                         REPI
      MT(3)=IABS(LM*(LM+1)+1-MA)                                       REPI
      V=ANGLI(LT,MT)*64.D0/((LL+LL+1)*(LR+LR+1))                       REPI
      IF(V.EQ.0.0D0) GO TO 38                                          REPI
      IF ((LL.EQ.LSA.AND.LR.EQ.LSB).OR.(LL.EQ.LSB.AND.LR.EQ.LSA)) GO TO REPI
     137                                                               REPI
      NLM=NL-LL                                                        REPI
      NLP=NL+LL+1                                                      REPI
      NRM=NR-LR                                                        REPI
      NRP=NR+LR+1                                                      REPI
      NCM=NC-LL-LR-1                                                   REPI
      SA=UF(NCP,NLM,NRM,WA,WB)-UF(NCP,NLM,NRP,WA,WB)-UF(NCP,NLP,NRM,WA, REPI
     1WB)                                                              REPI
      IF(NCM.GT.0) SA=SA+UF(NCP,NLP,NRP,WA,WB)                         REPI
      SUM=SA/S**NCP+VF(NCP,NLM,NRP,RA,AAM,ABM)+VF(NCP,NRM,NLP,RB,BBM,BAMREPI
     1)                                                                REPI
      IF(NCM.LE.0) GO TO 35                                            REPI
      SUM=SUM-VF(NCP,NLP,NRP,RA,AAM,ABM)-VF(NCP,NRP,NLP,RB,BBM,BAM)+FACTREPI
     1(NLP)*FACT(NRP)*FACT(NCM)/(CL**NLP*CR**NRP*CC**NCM)              REPI
      GO TO 36                                                         REPI
   35 NCM=-NCM                                                         REPI
      NCM1=NCM+1                                                       REPI
      NCM2=NCM+2                                                       REPI
      NCM3=NCM+3                                                       REPI
      W=FACT(NLP)*FACT(NRP)/(CL**NLP*CR**NRP)                          REPI
      SID1=(DLOG(RA*RB/(CC*S))+PSI(NCM1))*(-CC)**NCM/FACT(NCM1)        REPI
      SID2=FIDA(RA,NCM,WA,AAM,    1)                                   REPI
      SID3=FIDA(RB,NCM,WA,BBM,    1)                                   REPI
      SID4=FIDA( S,NCM,WA, WB,NCM1)                                    REPI
      SID5=FIDB(RA,NCM,    1,    1,WA,NCM3,NLP,AAM,1)                  REPI
      SID6=FIDB(RB,NCM,    1,    1,WA,NCM3,NRP,BBM,1)                  REPI
      SID7=FIDB( S,NCM,    1,NCM1,WA,NCM3,NRP, WB,1)                   REPI
      SID8=FIDB( S,NCM,NCM2, NLP,WA,    1,NRP, WB,2)                   REPI
      SID=SID1-SID2-SID3+SID4-SID5-SID6+SID7+SID8                      REPI
      SIDT=SID*W                                                       REPI
      SUM=SUM+SIDT                                                     REPI
   36 LSA=LL                                                           REPI
      LSB=LR                                                           REPI
   37 REPI=REPI+FC(IL,I)*FC(IR,J)*FC(IM,K)*V*SUM                       REPI
   38 CONTINUE                                                         REPI
   39 CONTINUE                                                         REPI
   40 CONTINUE                                                         REPI
      RETURN                                                           REPI
      END                                                              REPI
C     FUNCTIONS TO ASSIST REPI                                         ANGI
C     /CFACT/ FACT AND /CPSI/ PSI REQUIRED                             ANGI
      FUNCTION ANGLI(LT,MT)                                            ANGI
      IMPLICIT REAL*8(A-H,O-Z)                                         ANGI
      INTEGER*4 LT(3),MT(3)                                            ANGI
      COMMON /CFACT/ FACT(41)                                          ANGI
      COMMON /CPSI/ PSI(11)                                            ANGI
      ANGLI=0.0D0                                                      ANGI
```

```
      IST=LT(1)+LT(2)+LT(3)
      IF((-1)**IST.LT.0) RETURN                                         ANGI
      DO 1 I=1,3                                                        ANGI
      IA=I+1-3*(I/3)                                                    ANGI
      IB=I+2-3*(I/2)                                                    ANGI
      LU=LT(I)                                                          ANGI
      MU=MT(I)                                                          ANGI
      IF (MU.EQ.MT(IA)+MT(IB).AND.LU.LE.LT(IA)+LT(IB).AND.LU.GE.IABS(LT(ANGI
     1IA)-LT(IB))) GO TO 2                                              ANGI
    1 CONTINUE                                                          ANGI
      RETURN                                                            ANGI
    2 IF (LT(IA).GE.LT(IB)) GO TO 3                                     ANGI
      MC=IA                                                             ANGI
      IA=IB                                                             ANGI
      IB=MC                                                             ANGI
    3 LV=LT(IA)                                                         ANGI
      MV=MT(IA)                                                         ANGI
      LW=LT(IB)                                                         ANGI
      MW=MT(IB)                                                         ANGI
      IS=IST/2+1                                                        ANGI
      MA=MIN0(LW-MW,LU-MU)+1                                            ANGI
      X=-1.D0                                                           ANGI
      DO 4 I=1,MA                                                       ANGI
      X=-X                                                              ANGI
    4 ANGLI=ANGLI+X*FACT(LU+MU+I)*FACT(LV+LW-MU-I+2)/(FACT(I)*FACT(LU-MUANGI
     1-I+2)*FACT(LV-LW+MU+I)*FACT(LW-MW-I+2))                           ANGI
      ANGLI=ANGLI*FACT(LV+MV+1)*FACT(LW+MW+1)*FACT(IS)*FACT(IST-LW-LW+1)ANGI
     1*(-1.D0)**(IS-LV-MW)/(FACT(IS-LU)*FACT(IS-LV)*FACT(IS-LW)*        ANGI
     2FACT(IST+2)*FACT(LV-MV+1))                                        ANGI
      RETURN                                                            ANGI
      ENTRY UF(NCP,NLM,NRM,WA,WB)                                       ANGI
      UF=0.D0                                                           ANGI
      FK=WA                                                             ANGI
      DO 9 K=1,NLM                                                      ANGI
      UFA=0.D0                                                          ANGI
      FK=FK/WA                                                          ANGI
      FL=WB                                                             ANGI
      DO 8 L=1,NRM                                                      ANGI
      FL=FL/WB                                                          ANGI
    8 UFA=UFA+FL*FACT(NCP-NLM-NRM+K+L-2)/FACT(L)                        ANGI
    9 UF=UF+FK*UFA/FACT(K)                                              ANGI
      UF=UF*FACT(NLM)*FACT(NRM)*WA*WB/(FK*FL)                           ANGI
      RETURN                                                            ANGI
      ENTRY VF(NCP,NLM,NRP,PA,AAM,ABM)                                  ANGI
      VF=0.D0                                                           ANGI
      FK=AAM                                                            ANGI
      DO 10 K=1,NLM                                                     ANGI
      FK=FK/AAM                                                         ANGI
   10 VF=VF+FK*FACT(NCP-NLM-NRP+K-1)/FACT(K)                            ANGI
      VF=VF*FACT(NLM)*FACT(NRP)*AAM*ABM**NRP/(PA**NCP*FK)               ANGI
      RETURN                                                            ANGI
      ENTRY FIDA(V1,IV2,V3,V4,IV5)                                      ANGI
      Y=0.D0                                                            ANGI
      DO 12 L=1,IV5                                                     ANGI
      X=0.D0                                                            ANGI
      KUP=IV2-L+2                                                       ANGI
      DO 11 K=1,KUP                                                     ANGI
      X=X+FACT(KUP)*PSI(KUP+1-K)/(FACT(K)*FACT(KUP+1-K)*(-V4)**(K-1))   ANGI
   11 CONTINUE                                                          ANGI
      Y=Y+X/(FACT(L)*FACT(KUP)*(-V3)**(L-1))
```

```
12 CONTINUE
   FIDA=Y*(-V1)**IV2                                                   ANGI
   RETURN                                                              ANGI
   ENTRY FIDB(V1,IV2,IV3,IV4,V5,IV6,IV7,V8,IV9)                        ANGI
   Y=0.D0                                                              ANGI
   DO 14 L=IV3,IV4                                                     ANGI
   IF(IV9.EQ.1) KLO=IV6-L                                              ANGI
   IF(IV9.EQ.2) KLO=IV6                                                ANGI
   X=0.D0                                                              ANGI
   DO 13 K=KLO,IV7                                                     ANGI
   X=X+FACT(K+L-IV2-2)/(FACT(K)*V8**(K-1))                             ANGI
13 CONTINUE                                                            ANGI
   Y=Y+X/(V5**(L-1)*FACT(L))                                           ANGI
14 CONTINUE                                                            ANGI
   FIDB=Y*V1**IV2                                                      ANGI
   RETURN                                                              ANGI
   END                                                                 ANGI
C     NORMALIZATION FUNCTION (IMAGINARY)                               ENMI
   FUNCTION ENMI(N,L,M,C)                                              ENMI
   IMPLICIT REAL*8(A-H,O-Z)                                            ENMI
   REAL*8 CN(13)/2.D0,.66666666666666667,.88888888888888889D-1,        ENMI
  1.63492063492063492D-2,.28218694885361155D-3,                        ENMI
  2.85511196622307 7D-5,.18793669587320380-6,                          ENMI
  3.31322782645533 97D-8,.40944813915730670-10,                        ENMI
  4.43099804121821 74D-12,.37315847724521D-14,.27040469365594870-16,   ENMI
  5.16640288840366 1D-18/,CT(15)/.5D0,.75D0,1.5D0,.10416666666666667,  ENMI
  2.41666666666666667,2.5D0,.46111111111111111D-2,.29166666666666667D-1,ENMI
  3.29166666666666667,3.5D0,.11080714285714290-3,.33928571428571430-2,ENMI
  4.125D-1,.225D0,4.5D0/                                               ENMI
   ENMI=DSQRT(C**(N+N+1)*CT(((L+1)*(L+2))/2-IABS(M))*CN(N))            ENMI
   RETURN                                                              ENMI
   END                                                                 ENMI
C     SCHMIDT MATRIX MULTIPLICATION SUBROUTINE                         MULT
   SUBROUTINE MULTS(NB,H,EM,T)                                         MULT
   IMPLICIT REAL*8(A-H,O-Z)                                            MULT
   REAL*8 H(2),EM(2),T(2)                                              MULT
   DO 4  N=1,2                                                         MULT
   DO 4  I=1,NB                                                        MULT
   DO 4  J=1,NB                                                        MULT
   JQ=(J*(J-1))/2                                                      MULT
   IJ=JQ+I                                                             MULT
   B=0.0D0                                                             MULT
   MA=J                                                                MULT
   IF (N.EQ.2) MA=I                                                    MULT
   DO 2  K=1,MA                                                        MULT
   IK=MAX0(I,K)                                                        MULT
   IK=(IK*(IK-3))/2+I+K                                                MULT
   KJ=JQ+K                                                             MULT
   IF (N.EQ.2) GO TO 1                                                 MULT
   B=B+H(IK)*EM(KJ)                                                    MULT
   GO TO 2                                                             MULT
 1 B=B+EM(IK)*T(KJ)                                                    MULT
 2 CONTINUE                                                            MULT
   IF (N.EQ.2) GO TO 3                                                 MULT
   T(IJ)=B                                                             MULT
   GO TO 4                                                             MULT
 3 H(IJ)=B                                                             MULT
 4 CONTINUE                                                            MULT
   RETURN                                                              MULT
   END
```

```
C     SCHMIDT MATRIX-EIGENVECTOR MATRIX MULTIPLICATION SUBROUTINE        VMUL
      SUBROUTINE VMULT(MO,C,EM,ND)                                       VMUL
      IMPLICIT REAL*8(A-H,O-Z)                                           VMUL
      REAL*8 C(ND,ND),EM(2)                                              VMUL
      DO 2 I=1,MO                                                        VMUL
      DO 2 J=1,MO                                                        VMUL
      X=0.0D0                                                            VMUL
      DO 1 K=I,MO                                                        VMUL
    1 X=X+EM((K*(K-1))/2+I)*C(K,J)                                       VMUL
    2 C(I,J)=X                                                           VMUL
      RETURN                                                             VMUL
      END                                                                VMUL
C     SCHMIDT ORTHOGONALIZATION SUBROUTINE                               SOMS
      SUBROUTINE SOMS(MO,S,EM)                                           SOMS
      IMPLICIT REAL*8(A-H,O-Z)                                           SOMS
      REAL*8 CUT/1.D-6/,EM(2),S(2)                                       SOMS
      IER=0                                                              SOMS
      EM(1)=DSQRT(S(1))                                                  SOMS
      IF (MO.LT.2) GO TO 6                                               SOMS
      DO 5 I=2,MO                                                        SOMS
      IQ=(I*(I-1))/2                                                     SOMS
      IA=IQ+1                                                            SOMS
      EM(IA)=S(IA)/EM(1)                                                 SOMS
      DO 4 J=2,I                                                         SOMS
      JQ=(J*(J-1))/2                                                     SOMS
      IJ=IQ+J                                                            SOMS
      X=S(IJ)                                                            SOMS
      JM=J-1                                                             SOMS
      DO 1 K=1,JM                                                        SOMS
    1 X=X-EM(IQ+K)*EM(JQ+K)                                              SOMS
      IF (I.EQ.J) GO TO 2                                                SOMS
      EM(IJ)=X/EM(JQ+J)                                                  SOMS
      GO TO 4                                                            SOMS
    2 IF (X.GT.CUT.OR.IER.EQ.1) GO TO 3                                  SOMS
      WRITE (6,901) X                                                    SOMS
      IER=1                                                              SOMS
    3 EM(IJ)=DSQRT(X)                                                    SOMS
    4 CONTINUE                                                           SOMS
    5 CONTINUE                                                           SOMS
    6 DO 8 I=1,MO                                                        SOMS
      X=1.D0                                                             SOMS
      DO 8 J=I,MO                                                        SOMS
      JQ=(J*(J-1))/2                                                     SOMS
      Y=1.D0/EM(JQ+J)                                                    SOMS
      IF (I.EQ.J) GO TO 8                                                SOMS
      X=0.0D0                                                            SOMS
      JM=J-1                                                             SOMS
      DO 7 K=I,JM                                                        SOMS
    7 X=X-EM((K*(K-1))/2+I)*EM(JQ+K)                                     SOMS
    8 EM(JQ+I)=X*Y                                                       SOMS
      RETURN                                                             SOMS
  901 FORMAT ('0',131('*')/23X,'WARNING: DETERMINANT IN SOMS IS',D16.8,  SOMS
     1'. CHECK RESULT FOR LOSS IN PRECISION.'/1X,131('*')/)              SOMS
      END                                                                SOMS
      THE FOLLOWING ROUTINES ARE THE ROUTINES USED TO EDIT THIS THESIS   EDIT
C  THEFOLLOWING ROUTINES ARE THE ROUTINES USED TO EDIT THIS THESIS       EDIT
C THE FIRST PROGRAM READJUSTS INPUT TO THE PROPER THESIS SIZE            EDIT
      INTEGER*2 TE(80),BUF(80),BLANK/' '/,DOT/'.'/,STAR/'*'/,BAR/'|'/,   EDIT
     .COMMA/','/,PLUS/'+'/,DOLLAR/'$'/,HYPH/'-'/,QUER/'?'/,BLANKV(10)/   EDIT
     .10*' '/,ZAHL(10)/'1','2','3','4','5','6','7','8','9','0'/,LEPAR/   EDIT
     .'('/
```

```
      INTEGER ICOL/60/,LINEND/28/,IDENT/5/,PMARK/'****'/,BLAVEC(25)      EDIT
      LOGICAL JUMP                                                      EDIT
C   THE PROGRAM DOES:                                                   EDIT
C     A)READ A LINE WITH A MAXIMUM WITH OF 80 CHARACTERS FROM UNIT(1)   EDIT
C     B)LOOKS IF LINE IS TO BE EDITED( FIRST CHARACTER IF NO-EDIT)      EDIT
C     C)LOOKS IF A NEW PARAGRAPH IS TO BE STARTED(* FOLLOWED BY NEW LINEEDIT
C     D)ASKS FOR A STARTING PAGE NUMBER AND NUMBERS THE PAGES SUCESSIVELEDIT
C     E) $ STANDS FOR LITERAL NEXT                                      EDIT
900   FORMAT(' ENTER THE STARTING PAGE NUMBER IN 3-DIGIT FORM'/10X' I.E.EDIT
     . FIFTEEN AS 015')                                                 EDIT
901   FORMAT(I3)                                                        EDIT
902   FORMAT(57X,I3)                                                    EDIT
903   FORMAT(80A1)                                                      EDIT
904   FORMAT(' A WORD LONGER THAN TEN LETTERS WITHOUT A HYPHEN HAS BEEN EDIT
     .ENCOUNTERED'/10X,' CHANGE THIS WORD AND RESTART'/80A1)            EDIT
906   FORMAT(A4)                                                        EDIT
C   THE STARTING PAGE NO. IS ASKED FOR                                  EDIT
      WRITE(6,900)                                                      EDIT
      READ(5,901)IPAGE                                                  EDIT
      LINCNT=3                                                          EDIT
      WRITE(2,902)IPAGE                                                 EDIT
      WRITE(2,903)BLANKV                                                EDIT
      JB=1                                                              EDIT
1     READ(1,903,END=34)TE                                              EDIT
      JUMP=.FALSE.                                                      EDIT
      IBLAN=0                                                           EDIT
      IF(TE(1).NE.BAR)GOTO2                                             EDIT
      IF(JB.EQ.1)GOTO3                                                  EDIT
      JB=JB-1                                                           EDIT
      WRITE(2,903)(BUF(JA),JA=1,JB)                                     EDIT
      LINCNT=LINCNT+1                                                   EDIT
      JB=1                                                              EDIT
3     WRITE(2,903)(TE(JA),JA=2,61)                                      EDIT
      LINCNT=LINCNT+1                                                   EDIT
      IF(LINCNT.LT.LINEND)GOTO1                                         EDIT
      WRITE(2,906)PMARK                                                 EDIT
      IPAGE=IPAGE+1                                                     EDIT
      WRITE(2,902)IPAGE                                                 EDIT
      WRITE(2,903)BLANKV                                                EDIT
      LINCNT=3                                                          EDIT
      GOTO1                                                             EDIT
2     IF(TE(1).NE.QUER)GOTO4                                            EDIT
      IF(JB.EQ.1)GOTO5                                                  EDIT
      JB=JB-1                                                           EDIT
      WRITE(2,903)(BUF(JA),JA=1,JB)                                     EDIT
      JB=1                                                              EDIT
5     WRITE(2,906)PMARK                                                 EDIT
      LINCNT=3                                                          EDIT
      IPAGE=IPAGE+1                                                     EDIT
      WRITE(2,902)IPAGE                                                 EDIT
      WRITE(2,903)BLANKV                                                EDIT
      GOTO1                                                             EDIT
4     IF(TE(1).NE.PLUS)GOTO10                                           EDIT
      IF(JB.EQ.1)GOTO6                                                  EDIT
      JB=JB-1                                                           EDIT
      WRITE(2,903)(BUF(JA),JA=1,JB)                                     EDIT
      JB=1                                                              EDIT
      LINCNT=LINCNT+1                                                   EDIT
6     IFREE=0                                                           EDIT
      DO 70 JA=2,3
```

```
      DO 7 JC=1,10
      IF(TE(JA).NE.ZAHL(JC))GOTO7                                        EDIT
      IFREE=IFREE+(1-MOD(JA,2))*10*MOD(JC,10)+(JA-2)*MOD(JC,10)         EDIT
      GOTO70                                                             EDIT
7     CONTINUE                                                           EDIT
70    CONTINUE                                                           EDIT
      IF(IFREE+LINCNT.LE.LINEND)GOTO18                                   EDIT
      IPAGE=IPAGE+1                                                      EDIT
      WRITE(2,906)PMARK                                                  EDIT
      WRITE(2,902)IPAGE                                                  EDIT
      WRITE(2,903)BLANKV                                                 EDIT
      LINCNT=3                                                           EDIT
18    DO 8 JA=1,IFREE                                                    EDIT
      LINCNT=LINCNT+1                                                    EDIT
8     WRITE(2,903)BLANKV                                                 EDIT
      GOTO1                                                              EDIT
10    DO 11 JA=1,80                                                      EDIT
      IF(JUMP)GOTO14                                                     EDIT
      IF(TE(JA).NE.BLANK)GOTO12                                          EDIT
      IF(JB.EQ.1)GOTO11                                                  EDIT
      IBLAN=IBLAN+1                                                      EDIT
      IF(IBLAN.GT.1)GOTO11                                               EDIT
      GOTO13                                                             EDIT
12    IBLAN=0                                                            EDIT
      IF(TE(JA).NE.DOLLAR)GOTO15                                         EDIT
      JUMP=.TRUE.                                                        EDIT
      GOTO11                                                             EDIT
15    IF(TE(JA).EQ.HYPH.AND.ICOL-10.GT.JB)GOTO11                         EDIT
      IF(TE(JA).EQ.STAR)GOTO16                                           EDIT
13    IF(ICOL-10.GT.JB)GOTO14                                            EDIT
      IF(TE(JA).EQ.COMMA.OR.TE(JA).EQ.HYPH.OR.TE(JA).EQ.DOT.OR.TE(       EDIT
     .JA).EQ.BLANK)GOTO17                                                EDIT
      GOTO21                                                             EDIT
17    NOBLA=ICOL-JB                                                      EDIT
      IST=1                                                              EDIT
170   DO 19 JC=IST,NOBLA                                                 EDIT
      IF(JA+JC.GT.80)GOTO190                                             EDIT
      IF(TE(JA+JC).EQ.BLANK)GOTO141                                      EDIT
      IF(TE(JA+JC).EQ.HYPH)GOTO140                                       EDIT
      IF(TE(JA+JC).EQ.DOT)GOTO140                                        EDIT
      IF(TE(JA+JC).EQ.COMMA)GOTO140                                      EDIT
19    CONTINUE                                                           EDIT
190   IF(TE(JA).NE.BLANK)BUF(JB)=TE(JA)                                  EDIT
      IF(TE(JA).EQ.BLANK)NOBLA=NOBLA+1                                   EDIT
      GOTO20                                                             EDIT
141   IF(JC.NE.IST)GOTO140                                               EDIT
      IST=IST+1                                                          EDIT
      GOTO170                                                            EDIT
21    IF(JB.NE.ICOL)GOTO14                                               EDIT
      IF(TE(JA+1).NE.BLANK)GOTO22                                        EDIT
      BUF(JB)=TE(JA)                                                     EDIT
      GOTO23                                                             EDIT
22    WRITE (6,904)TE                                                    EDIT
      STOP                                                               EDIT
140   IF(TE(JA).EQ.HYPH)GOTO11                                           EDIT
14    BUF(JB)=TE(JA)                                                     EDIT
      JUMP=.FALSE.                                                       EDIT
      JB=JB+1                                                            EDIT
      IF(JB.LT.ICOL-10)GOTO11                                            EDIT
      IF(TE(JA).EQ.HYPH.OR.TE(JA).EQ.BLANK)GOTO24
```

```
      IF(TE(JA).NE.DOT.AND.TE(JA).NE.COMMA)GOTO11                       EDIT
24    NOBLA=ICOL-JB+1                                                   EDIT
      IST=1                                                             EDIT
240   DO 25 JC=IST,NOBLA                                                EDIT
      IF(JA+JC.GT.80)GOTO20                                             EDIT
      IF(TE(JA+JC).EQ.BLANK)GOTO250                                     EDIT
      IF(TE(JA+JC).EQ.HYPH)GOTO11                                       EDIT
      IF(TE(JA+JC).EQ.DOT.OR.TE(JA+JC).EQ.COMMA)GOTO11                  EDIT
25    CONTINUE                                                          EDIT
      IF(TE(JA+NOBLA+1).EQ.BLANK)GOTO11                                 EDIT
      GOTO20                                                            EDIT
250   IF(JC.NE.1)GOTO11                                                 EDIT
      IST=IST+1                                                         EDIT
      GOTO240                                                           EDIT
C   A NEW PARAGRAPH IS STARTED                                          EDIT
16    JB=JB-1                                                           EDIT
      IF(JB.EQ.0)GOTO260                                                EDIT
      WRITE(2,903)(BUF(JC),JC=1,JB)                                     EDIT
      LINCNT=LINCNT+1                                                   EDIT
      IF(LINCNT.NE.LINEND)GOTO260                                       EDIT
      IPAGE=IPAGE+1                                                     EDIT
      WRITE(2,906)PMARK                                                 EDIT
      WRITE(2,902)IPAGE                                                 EDIT
      WRITE(2,903)BLANKV                                                EDIT
      LINCNT=3                                                          EDIT
260   DO 26 JC=1,IDENT                                                  EDIT
26    BUF(JC)=BLANK                                                     EDIT
      JB=IDENT+1                                                        EDIT
      GOTO11                                                            EDIT
C   BUF IS RIGHT JUSTIFIED, I.E. BLANKS ARE REMOVED                     EDIT
20    IF(NOBLA.EQ.0)GOTO23                                              EDIT
27    JD=1                                                              EDIT
      JC=IDENT+1                                                        EDIT
28    IF(BUF(JC).NE.BLANK)GOTO29                                        EDIT
      IF(BUF(JC+1).EQ.LEPAR)GOTO29                                      EDIT
      BLAVEC(JD)=JC                                                     EDIT
      JD=JD+1                                                           EDIT
29    JC=JC+1                                                           EDIT
      IF(JB-1.GT.JC)GOTO28                                              EDIT
30    JE=1                                                              EDIT
31    JF=(JE+1)/2                                                       EDIT
      IF(MOD(JE,2).EQ.0)JF=JD-JF                                        EDIT
      LIMG=ICOL-NOBLA+1                                                 EDIT
      LIM=LIMG-BLAVEC(JF)                                               EDIT
      DO 32 JG=1,LIM                                                    EDIT
32    BUF(LIMG+1-JG)=BUF(LIMG-JG)                                       EDIT
      LIMJ=JD+1-JF                                                      EDIT
      NOBLA=NOBLA-1                                                     EDIT
      IF(NOBLA.EQ.0)GOTO23                                              EDIT
      DO 33 JG=1,LIMJ                                                   EDIT
33    BLAVEC(JD-JG+1)=BLAVEC(JD-JG+1)+1                                 EDIT
      JE=JE+1                                                           EDIT
      IF(JE.LE.JD)GOTO31                                                EDIT
      GOTO30                                                            EDIT
23    WRITE(2,903)(BUF(JG),JG=1,ICOL)                                   EDIT
      LINCNT=LINCNT+1                                                   EDIT
      IF(LINCNT.EQ.LINEND)GOTO9                                         EDIT
      JB=1                                                              EDIT
      GOTO11                                                            EDIT
9     LINCNT=3
```

```
      JB=1                                                               EDIT
      WRITE(2,906)PMARK                                                  EDIT
      IPAGE=IPAGE+1                                                      EDIT
      WRITE(2,902)IPAGE                                                  EDIT
      WRITE(2,903)BLANKV                                                 EDIT
11    CONTINUE                                                           EDIT
      GOTO1                                                              EDIT
34    LIMB=JB-1                                                          EDIT
      WRITE(2,903)(BUF(JG),JG=1,LIMB)                                    EDIT
      STOP                                                               EDIT
C     DEBUG UNIT(9),TRACE,SUBCHK,INIT(BUF,JB,NOBLA,LINCNT)               EDIT
C     AT1                                                                EDIT
C     TRACE ON                                                           EDIT
      END                                                                EDIT
C   THIS IS THE PRINT VERSION FOR TERMINAL USE                           PRTL
C   ISTA,IEND ARE THE FIRST AND LAST LINE NO. OF THE FILE TO BE          PRTL
C                                                                        PRTL
C   EDITED. PLEASE NOTE THAT THE FILE MUST BE NUMBERED IN INCREMENTS OF  PRTL
      INTEGER TE(15),BLAV(2)/2*'    '/,PMARK/'****'/,CONTIN,BLANK/' '/   PRTL
      READ(5,904)ISTA,IEND                                               PRTL
3     DO 1 JA=1,40                                                       PRTL
      IF(ISTA+JA-1.GT.IEND)GOTO2                                         PRTL
      READ(1'(ISTA+JA-1)*1000,903)TE                                     PRTL
      IF(TE(1).EQ.PMARK)GOTO2                                            PRTL
      WRITE(2,900)TE                                                     PRTL
1     WRITE(2,900)BLAV                                                   PRTL
2     READ(5,901)CONTIN                                                  PRTL
      IF(CONTIN.EQ.BLANK)GOTO3                                           PRTL
      ISTA=ISTA+JA                                                       PRTL
      GOTO3                                                              PRTL
900   FORMAT(' ',15A4)                                                   PRTL
901   FORMAT(10A1)                                                       PRTL
902   FORMAT('1')                                                        PRTL
903   FORMAT(15A4)                                                       PRTL
904   FORMAT(2I4)                                                        PRTL
      END                                                                PRTL
C   LINE-PRINTER VERSION OF THE PRINT ROUTINE                            PRTL
      INTEGER TE(15),BLAV(4)/4*'    '/,PMARK/'****'/                     PRTL
      WRITE(2,901)                                                       PRTL
      WRITE(2,902)                                                       PRTL
3     DO 1 JA=1,40                                                       PRTL
      READ(1,903,END=4)TE                                                PRTL
      IF(TE(1).EQ.PMARK)GOTO2                                            PRTL
1     WRITE(2,900)BLAV,TE                                                PRTL
2     WRITE(2,901)                                                       PRTL
      WRITE(2,902)                                                       PRTL
      GOTO3                                                              PRTL
4     WRITE(2,901)                                                       PRTL
      STOP                                                               PRTL
900   FORMAT('0',19A4)                                                   PRTL
901   FORMAT('1')                                                        PRTL
902   FORMAT(//)                                                         PRTL
903   FORMAT(15A4)                                                       PRTL
      END
```